

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Adaptation des entrées du simulateur de gestion pour le temps de partage

Docquier, Marie-Paule; Goffart, Marcelle

*Award date:*  
1980

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES  
UNIVERSITAIRES  
N.D. DE LA PAIX  
NAMUR

---

INSTITUT D'INFORMATIQUE

ADAPTATION DES ENTREES  
DU SIMULATEUR DE GESTION  
POUR LE TEMPS PARTAGE

Marie-Paule DOCQUIER  
Marcelle GOFFART

Mémoire présenté en vue  
de l'obtention du grade  
de Licencié et Maître  
en Informatique

ANNEE ACADEMIQUE 1979-1980



LBS 3590062

77163

Nous tenons à remercier  
Monsieur R. GIGOT, directeur du mémoire,  
pour l'intérêt qu'il a porté à  
l'évolution de ce travail.

Nous remercions également  
tous ceux qui nous ont apporté  
leurs conseils judicieux et leurs critiques.

## TABLE DES MATIERES

	PAGE
1. Introduction : Objet du mémoire	1
2. Description du simulateur de gestion	
2.1 Généralités	3
2.2 Découpe du système en ses différentes applications	4
3. Etude du système actuel de mise à jour de la banque de données	
3.1 Description des données	7
3.2 Description des traitements	19
4. Critique de l'existant	29
5. Etude du système prévu de mise à jour de la banque de données	30
5.1 Le gestionnaire de grilles	
5.1.1 Généralités	31
5.1.2 Le système DAC	32
5.1.3 Programme MAIN	36
5.1.4 Routines du gestionnaire de grilles	46
5.1.5 Description des grilles	70
5.2 Description des données	74
5.3 Description des traitements	
5.3.1 Description générale	76
5.3.2 Graphe général	78
5.3.3 Description d'une transaction dans le système proposé	79
5.3.4 Algorithmes des programmes d'application	131
6. Conclusions	138

Annexes

## INTRODUCTION

=====

## 1. OBJET DU MEMOIRE

=====

Notre mémoire a pour but l'adaptation des entrées du simulateur de gestion pour le temps partagé. Le simulateur de gestion est un outil d'aide à la gestion. Il permet, à partir de ressources centralisées dans une banque de données, de générer un programme linéaire qui décrit des relations et des contraintes de transformation de l'entreprise. Ce modèle mathématique est capable de définir le programme d'activité de cette entreprise. Dans le présent mémoire, nous nous intéressons à l'application de mise à jour de la banque de données. Nous allons réaliser l'introduction des données en temps partagé, par écran.

Ce système est destiné à remplacer un système batch d'entrées par cartes, ce dernier étant fort lourd actuellement. Cette lourdeur est due à la longueur du traitement et à l'enchaînement de nombreuses phases. Les différentes cartes correspondant à un article sont d'abord triées et regroupées par article. Le premier contrôle, effectué à ce niveau, vérifie sommairement chaque article zone par zone et attribue des valeurs par défaut. On réalise ensuite la mise à jour, par fonction (création, modification puis suppression), de la banque de données. C'est seulement dans ce programme de mise à jour qu'une seconde partie des contrôles peut s'effectuer. Il s'agit de contrôles de cohérence des zones d'un article et de cet article par rapport aux autres.

Le système en temps partagé que nous envisageons a pour objectif principal d'alléger la procédure de mise à jour tout en fournissant à l'utilisateur une manipulation aisée des données. Cet objectif a été atteint grâce à un gestionnaire de grilles que nous avons élaboré.

A chaque type d'article correspond un dessin d'écran (une grille) que nous avons voulu le plus clair possible. L'utilisateur réalise ses mises à jour de manière ponctuelle, article par article. A chaque transaction, l'utilisateur fait apparaître sur l'écran la grille formatée de l'article concerné.

En cas de modification ou de suppression, le contenu de l'article à traiter est visualisé sur l'écran. Un dialogue est établi entre l'utilisateur et la banque de données en vue de lui permettre de corriger ses erreurs au fur et à mesure, de prendre connaissance des valeurs par défaut.

Le gestionnaire de grilles assure les contrôles syntaxiques et de fourchettes. Les programmes d'application s'en trouvent fortement allégés. Ils n'assurent plus que les vérifications de cohérence.



## 2. LE SIMULATEUR DE GESTION

### 2.1 Généralités

Le simulateur de gestion budgétaire est un outil d'aide à la décision dans le domaine de la planification économique et budgétaire. Etant donné un plan de vente ( en prix et en quantité) ainsi que le prix et les disponibilités des ressources, l'outil doit déterminer le plan d'activité optimal, en quantité et en valeur, de l'entreprise. Cet outil est utilisé dans le but de répondre à certains problèmes de gestion en testant rapidement différentes hypothèses.

Si on pose les problèmes de gestion dans le contexte fonctionnel (problèmes de production, de vente ou d'approvisionnement ...), le simulateur apporte une réponse à ces problèmes en fournissant les différents budgets (budget de production, des ventes ou des approvisionnements ...).

De même le simulateur résout les problèmes de gestion suivant leur portée dans le temps :

- à un problème à long terme, de choix des investissements, il présente un budget qui traduit une politique des investissements, une politique générale de vente ....
- à moyen terme, il définit un plan d'activité
- il fournit, pour les problèmes à court terme, un budget qui est un "référentiel" par rapport auquel s'apprécie toute une série de décisions, par exemple le fait d'accepter ou de rejeter une commande.

Le simulateur, comme système informatique, est basé sur la représentation par deux modèles "duals" de la même réalités :

- un modèle mathématique qui est capable de définir le programme d'activité de l'entreprise, tout en tenant compte des équilibres internes de l'entreprise, du coût et des disponibilités des ressources et d'une demande potentielle connue en quantité et en valeur

./...

- un modèle de base de données qui jouit des propriétés suivantes :
  - . posséder une structure suffisamment riche pour représenter la totalité des relations internes de l'entreprise et suffisamment générale pour supporter sans trop de distorsion ;
  - . être apte à supporter les différents programmes d'application avec un maximum d'efficacité et de fiabilité.

## 2.2 Découpe du système en ses différentes applications

### 1. Mise à jour de la banque de données

La structure de banque de données étant définie, cette application permet à partir de commandes standardisées (des fonctions d'accès à la banque de données, implémentées dans le modèle IDS) d'effectuer toutes les mises à jour du contenu de la banque de données.

La mise à jour est donc aussi bien la création ou la modification que la suppression d'une entité (article IDS).

Les modifications du contenu d'une entité portent uniquement sur des données exogènes au système. Par exemple, la procédure de modification permet de changer le coût d'une ressource mais ne permet pas de changer le prix de revient d'un centre de frais, ce prix étant lui-même un des résultats du système.

L'application de mise à jour groupe en plus toute une série de contrôles dus à la structure de la banque de données et de contrôles ponctuels sur la validité des informations.

### 2. Génération du modèle mathématique

Le modèle mathématique choisi est un programme linéaire qui est généré automatiquement à partir du contenu de la banque de données. Cette application est paramétrée de telle façon que le modèle généré corresponde à la totalité de l'entreprise, ou bien à un sous-ensemble de secteurs de cette entreprise. Cette application permet aussi de détecter certaines erreurs dans la logique du système, erreurs que des contrôles ponctuels ne pouvaient détecter.

Si le modèle mathématique semble vraisemblable, la banque de données est verrouillée afin de la protéger de mises à jour intempestives qui risqueraient de rendre la banque de données incompatible avec les résultats qu'on devra réinsérer.

./...



### 3. Optimisation du modèle mathématique

L'optimisation du programme linéaire généré en 2. s'effectue à l'aide d'un software fourni par le constructeur, le MPS (Mathematical Programming System).

Si l'on obtient une solution optimale satisfaisante, on insère le programme d'activité obtenu dans la banque de données.

Sinon, il faut déverrouiller la banque de données et effectuer les corrections nécessaires.

### 4. Calcul des prix de revient

Un programme d'activité satisfaisant ayant été versé dans la banque de données, cette application sert à valoriser ce programme d'activité. La valorisation des résultats porte principalement sur l'élaboration des prix de revient des différents postes et sur l'agrégation de certains résultats pour en faire différentes synthèses.

### 5. Edition des budgets

L'édition des budgets est une application paramétrée qui permet à l'utilisateur de choisir entre dix-sept éditions qui couvrent à la fois des éditions détaillées et des éditions de synthèse. D'autre part, l'utilisateur, peut, à l'intérieur de chaque édition, sélectionner les sous-ensembles des éléments qu'il désire voir éditer.

### 6. Vidage pour mise à jour

Cette application se déroule en trois phases chronologiquement ordonnées et distinctes :

#### Phase 1 : Editions paramétrées des données significatives du système

Ces éditions ont été conçues de telle façon qu'une gamme relativement large d'utilisateurs puissent les comprendre et corriger éventuellement certaines données sur le listing proprement dit.

#### Phase 2 : Distribution et correction des états imprimés par les utilisateurs

#### Phase 3 : Mise à jour de la banque de données

Après la collecte des listings corrigés par les utilisateurs, on effectue l'encodage sous forme simplifiée des mises à jour à effectuer.

./...



A partir des mises à jour encodées sous forme simplifiées, des commandes de mise à jour exigée par l'application de mise à jour sont générées automatiquement.

Ensuite on met à jour la banque de données.

## 7. Les utilitaires

### 1. Déblocage de la banque de données

Après analyse des résultats, tant au niveau de l'optimisation qu'après les éditions budgétaires, si les résultats ne sont pas satisfaisants, il faut pouvoir déverrouiller la banque de données avant d'effectuer les mises à jour de correction.

### 2. Restructuration de la banque de données

Si pour un motif quelconque (par exemple saturation partielle ou totale de la banque de données), l'implantation physique de la banque de données n'est plus satisfaisante, on peut, à partir de la banque de données existante, redéfinir et recréer une nouvelle banque de données.

Cette application ne permet ni de changer les relations entre des entrées, ni de changer le nombre ou le contenu des entités ; elle permet de créer une implantation physique différente d'une banque de données fonctionnellement identique à la première.

### 3. ETUDE DU SYSTEME ACTUEL DE MISE A JOUR DE LA BANQUE DE DONNEES

=====

L'entrée des données se fait actuellement par cartes.  
Plusieurs cartes sont nécessaires pour décrire un article.

#### 3.1 Description des données

-----

Les données du système sont représentées à trois niveaux.

##### 1. Graphe d'activité

Les données sont représentées dans le graphe d'activité qui est un modèle économique-comptable de l'entreprise. Le graphe d'activité est une structuration de l'entreprise. Il représente les centres de frais, les ressources, les produits et les stades de transformation. De plus, il schématise les relations entre ces éléments et quantifie ces relations.

Ces relations sont soit des relations de transformation, soit des relations de consommation entre ressources et produits, soit des relations de transfert de produits entre usines.

Les standards de consommation quantifient les relations. Ces standards sont déduits de la comptabilité analytique.

##### 2. Modèles analytiques

Les modèles analytiques, un modèle d'optimisation et un modèle de calcul des prix de revient, simulent le comportement du graphe d'activité en fonction des sollicitations extérieures.

##### Modèle d'optimisation

C'est un programme linéaire décrivant :

- des variables : - le niveau d'activité des centres de frais
- le niveau d'approvisionnement des ressources
- le niveau de production des produits
- le niveau de transfert des produits entre usines
- le niveau de vente des produits

./...

- des équations d'équilibre correspondant aux valeurs du graphe d'activité
- des contraintes relatives aux variables
- des fonctions objectifs :
  - fonction de maximisation du chiffre d'affaires
  - fonction de minimisation des dépenses
  - fonction de maximisation de la marge bénéficiaire

Les résultats de ce modèle d'optimisation sont les budgets en quantité.

#### Modèle de calcul des prix de revient

Ce modèle valorise les budgets en quantité qui constituent le plan d'activité.

C'est donc ce modèle qui détermine les budgets en valeur :

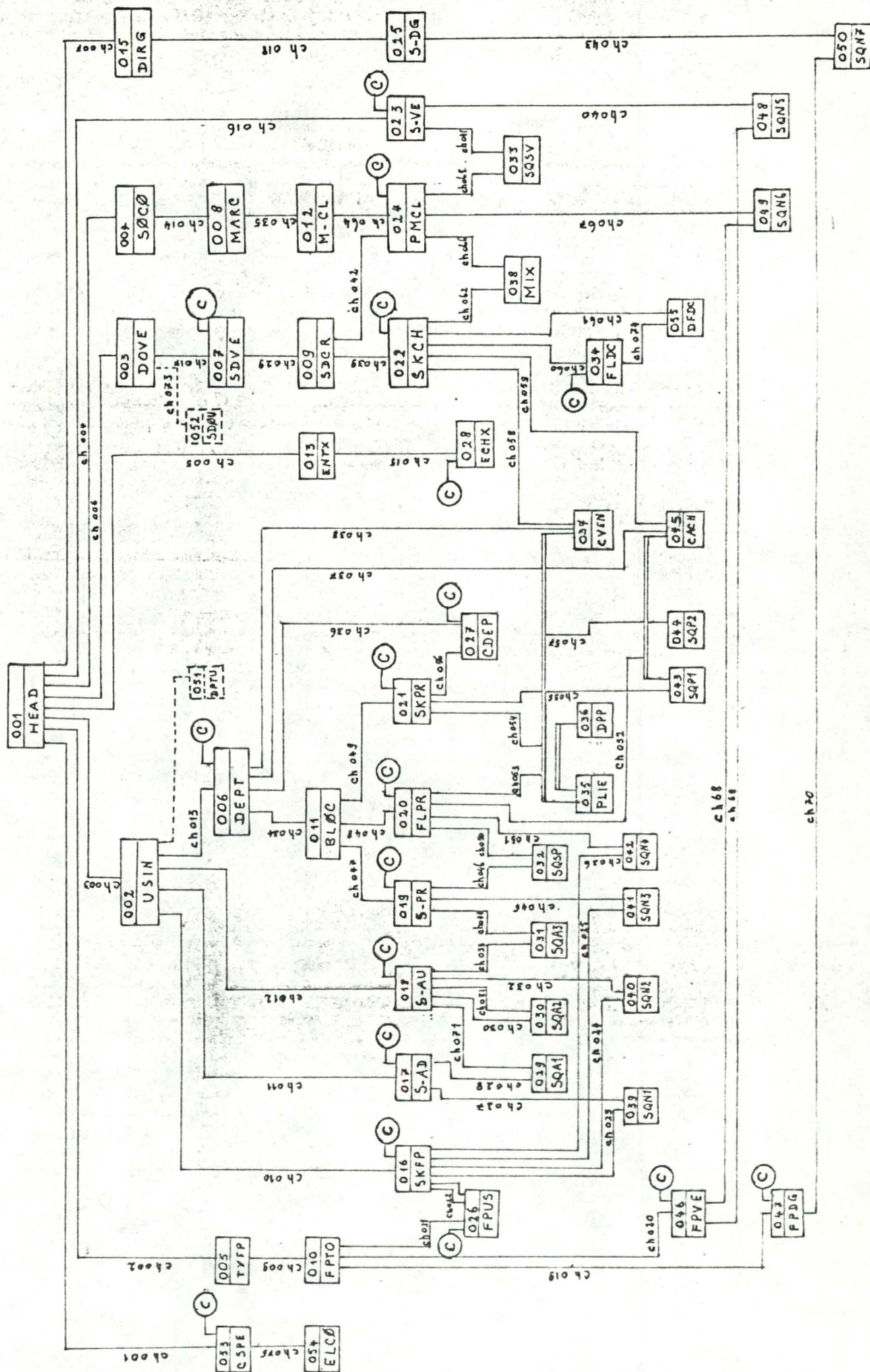
- budget des approvisionnements
- prix de revient complet et marginal de l'unité d'activité des différents postes de travail
- prix de revient des produits
- chiffre d'affaires et marge brute par marché, par client, par produit-marché-client

### 3. Modèle de banque de données

Le modèle de banque de données du simulateur est le modèle IDS - Integrated Data Stored - du software de la série 66 d'Honeywell Bull.



MODELE I.D.S.



## a) Modèle IDS

-----

Ce modèle de banque de données décrit les éléments du graphe d'activité et reflète, par sa structure, les relations entre ces éléments. Il décrit en fait l'entièreté de l'entreprise avec ses relations internes. Chaque élément du graphe d'activité correspond à un type dans la banque de données. Dans le modèle IDS, les relations inter-articles sont représentées par des chaînes. Ces chaînes sont composées d'un article appelé maître et d'articles appelés détails. Tous les articles d'une même chaîne forment une boucle, le dernier détail étant relié au maître. Le maître sert de point d'entrée dans la chaîne. C'est pourquoi une chaîne cesse d'exister (logiquement) lorsque son maître est supprimé : ses détails et les détails de ces derniers sont également supprimés pour toutes les chaînes auxquelles ils appartiennent.

Les articles décrits dans la banque de données sont soit des articles codés, soit des articles relations.

Pour les articles codés, l'identifiant est le code complet en ce sens qu'il reprend tous les codes de ses maîtres.

Exemple : le code de la section de production est composé de différents codes : numéro usine / numéro département / numéro bloc / numéro section

Pour les articles relations, l'identifiant est composé du code de l'émetteur et de celui du récepteur (deux fois quatorze caractères) c'est-à-dire les codes des deux articles mis en relation par l'article recherché.

Méthode d'accès pour les articles codés :

- si l'article est calculé : la recherche de cet article se fait directement
- si l'article est non calculé : il est retrouvé en descendant dans la structure, recherchant successivement les maîtres dont le code est repris dans le sien

Méthode d'accès pour les articles relations : on recherche tout d'abord l'un des deux maîtres sur base de son code. Ensuite, on recherche son premier article relation qui en est détail. A partir de celui-ci, on remonte au deuxième maître pour voir si l'occurrence de ce maître a pour identifiant celui spécifié.

Si oui : on teste le type de relation de l'article relation

./...



courant. En effet, on peut avoir plusieurs articles relations pour un seul émetteur et un seul récepteur.  
Par exemple, deux mêmes occurrences d'articles peuvent être reliées par deux standards différents, un standard fixe et un standard variable.

Si non : passer à l'article relation suivant.

Le langage de manipulation de la banque de données IDS pour les programmes d'application est, dans notre application, une extension du COBOL 68. Toutes les fonctions d'accès aux données implémentées dans le mode IDS sont réalisées dans ce langage.

b) Description du contenu de la banque de données

La banque de données est constituée de 54 types d'articles différents.

Chaque article est désigné par un code alphanumérique en quatre positions, appelé générique. A chaque article est attribué un numéro unique appelé numéro IDS, utilisé dans les accès à la banque de données. Les articles relations sont marqués de "R" ; les autres sont des articles codés.

Numéro	Générique	Types d'art.	Désignation
001	HEAD		Article de tête
002	USIN		Usine
003	DOVE		Domaine de vente
004	SOCO		Société de commercialisation
005	TYFP		Type de fourniture primaire
006	DEPT		Département
007	SDVE		Sous-domaine de vente
008	MARC		Marché
009	SDCR		Sous-domaine-critère
010	FPTO		Fourniture primaire pour l'entité
011	BLOC		Bloc
012	M-CL		Marché-client
013	ENTX		Entité extérieure

./...

Numéro	Générique	Types d'art.	Désignation
015	DIRG		Direction générale
016	SKFP		Stock (réel ou fictif) de fourniture primaire
017	S-AD		Section administrative
018	S-AU		Section auxiliaire
019	S-PR		Section de production
020	FLPR		Flux de produit
021	SKPR		Stock (réel ou fictif) de produit
022	SKCH		Stock d'échange
023	S-VE		Section de vente
024	PMCL		Produit-marché-client
025	S-DG		Section administrative générale
026	FPUS		Fourniture primaire pour usine
027	CDEP		Cession inter-département
028	ECHX		Echange avec l'extérieur
029	SQA1	R	Standard de consommation d'une section auxiliaire par une section adminis- trative d'usine
030	SQA2	R	Standard de consommation d'une section auxiliaire par une section auxiliaire (prestations réciproques et auto-consommations)
031	SQA3	R	Standard de consommation d'une section auxiliaire par une section de produc- tion
032	SQSP	R	Standard de consommation d'une section de produc- tion par un flux de produit
033	SQSV	R	Standard de consommation d'une section de vente par un produit marché-client
034	FLDC		Flux de déclassement

./...

Numéro	Générique	Types d'art.	Désignation
O35	PLIE	R	Transfert non intégral d'un flux de produit à un stock de produit
O36	DPP	R	Transfert automatique et intégral d'un flux de produit à un stock de produit
O37	CVEN	R	Standard de consommation d'un flux de produit par un stock d'échange
O38	MIX	R	Standard de consommation d'un stock d'échange par un produit-marché-client
O39	SQN1	R	Standard de consommation d'une fourniture primaire par une section administrative
O40	SQN2	R	Standard de consommation d'une fourniture primaire par une section auxiliaire
O41	SQN3	R	Standard de consommation d'une fourniture primaire par une section de production
O42	SQN4	R	Standard de consommation d'une fourniture primaire par un flux de produit
O43	SQP1	R	Standard de consommation d'un stock de produit par un flux de produit
O44	SQP2	R	Standard de consommation d'un produit cédé par un département par un produit d'un autre département
O45	CACH	R	Standard de consommation d'un stock d'échange par un flux de produit
O46	FPVE		Fourniture primaire de vente
O47	FPDG		Fourniture primaire de direction générale

./...



Numéro	Générique	Type d'art.	Désignation
048	SQN5	R	Standard de consommation d'une fourniture primaire de vente par une section de vente
049	SQN6	R	Standard de consommation d'une fourniture primaire de vente par un produit- marché-client
050	SQN7	R	Standard de consommation d'une fourniture primaire de direction générale par une section administrative générale
051	DPTU		Page-range d'un département
052	SDOU		Page-range d'un sous-domaine de vente
053	CSPE		Contrainte spéciale
054	ELCO		Elément de la contrainte spéciale

Un article codé est décrit de la façon suivante :

- une zone de codification en quatorze positions  
alphanumériques
- un ensemble de zones de données
- un ensemble de zones calculées

Un article relation est décrit de la façon suivante :

- un ensemble de zones de données où est repris le  
type de relation
- un ensemble de zones calculées

## b.1) La codification

La codification est hiérarchisée. En effet, la codification d'un élément reprend la codification de l'élément supérieur. Les codifications se subdivisent en trois grands schémas.

I. CODIFICATION DES RESSOURCES

	Usine	Département	Type de ressource		Nature de la ressource			FOURNISSEUR							
Type de ressources			x	x											
Fourniture primaire pour l'entité			x	x	x	x	x								
Fourniture primaire pour une usine	x	(x)	x	x	x	x	x	x	x	x	x	x	x	x	x
Stock de fourniture primaire	x	(x)	x	x	x	x	x								
Fourniture primaire de vente	0		x	x	x	x	x	x	x	x	x	x	x	x	x
Fourniture primaire de direction générale	1		x	x	x	x	x	x	x	x	x	x	x	x	x

II. CODIFICATION DE LA PRODUCTION

	Usine	Département	Bloc		Section			PRODUIT							
Usine	x														
Département	x	x													
Bloc	x	x	x	x											
Section de production	x	x	x	x	x	x	x								
Section auxiliaire	x				x	x	x								
Section administrative	x				x	x	x								
Flux de produit	x	x	x	x	(x	x	x)	x	x	x	x	x	x	x	x
Stock de produit	x	x	x	x				x	x	x	x	x	x	x	x
Cession inter-départ	x	x	x	x			x	x	x	x	x	x	x	x	x

## III. CODIFICATION DE LA VENTE

	⌒ Produit							⌒ Commercial						
	Domaine		Sous domaine	s-s domaine	Autres critères de produits			Société de commercial.	Marché			Client		
Domaine de vente	x	x												
Sous-domaine de vente	x	x	x											
Sous-domaine critère	x	x	x	x										
Sté de commercialisation								x						
Marché								x	x	x	x			
Marché-client								x	x	x	x	x	x	x
Produit-marché-client	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Echange avec l'extérieur	x	x	x	x	x	x	x	fournisseur						
Stock d'échange	x	x	x	x	x	x	x							



## b.2) Les zones de données

- - - - -

Les données peuvent être subdivisées entre deux grandes classes.

I. DONNEES ESSENTIELLES

- le prix d'achat de toutes les ressources et éventuellement des contraintes de disponibilités
- tous les standards de consommation
- les rendements des produits aux différents stades
- les contraintes de capacité des sections
- le prix et les "fourchettes" de ventes pour tous les produits à vendre
- certains prix d'ordre (pour les variations de stock et pour les produits recyclés)
- les prix de cession et les contraintes de cession si l'on effectue des optimisations partielles

II. DONNEES FACULTATIVES

- des libellés
- des libellés d'unité
- des bornes de vraisemblances
  - de prix d'achat des ressources
  - de prix de vente des produits
  - de prix de revient des produits
  - de standards
  - de rendement

## b.3) Les zones calculées

- - - - -

Les zones résultats sont garnies uniquement par le programme linéaire ; elles ne peuvent être accessibles à l'utilisateur et ainsi ne sont pas concernées par l'application de mise à jour.

./...

### 3.2 Description des traitements

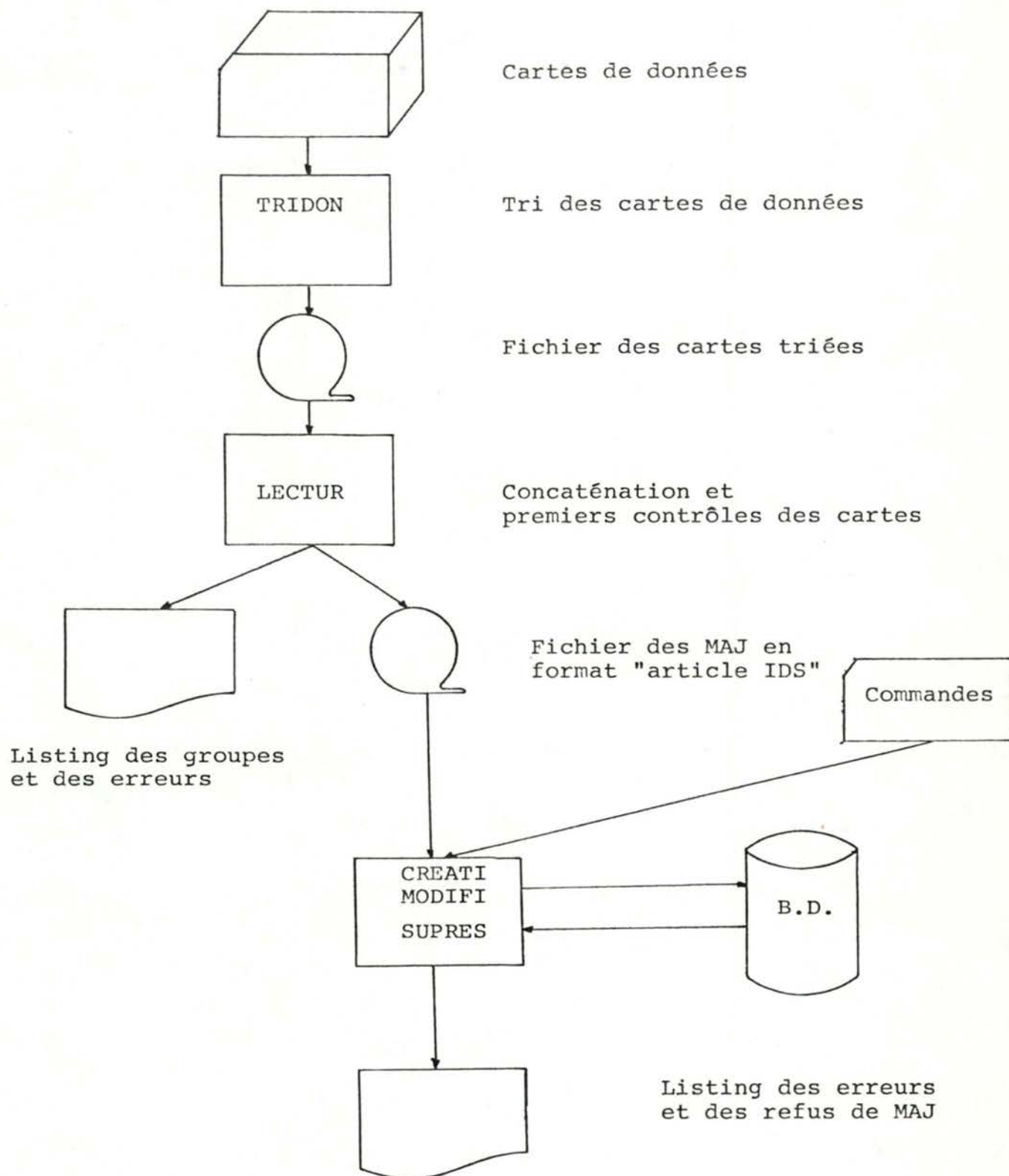
-----

#### Graphe de l'application de mise à jour de la banque de données

Cette application est composée des 5 phases suivantes :

1. Tri des cartes de données - TRIDON -
2. Procédure de concaténation - LECTUR -
3. Procédure de création - CREATI -
4. Procédure de modification - MODIFI -
5. Procédure de suppression - SUPRES -

./...

GRAPHE DE L'APPLICATION

## 1. Tri des cartes de données TRIDON

Les cartes données d'entrée sont triées sur les clés suivantes :

- code fonction
- code carte
- générique
- codification
- code séquence (car plusieurs cartes sont nécessaires pour décrire un article)

Le tri est ascendant. Il s'effectue de façon à ce que :

- toutes les cartes qui se rapportent à un même article IDS se suivent en ordre croissant sur leur numéro de séquence
- l'ordre d'apparition des groupes ainsi formés soit le même que celui exigé par la hiérarchie de la banque de données (un maître doit être avant son détail)

## 2. Procédure de concaténation LECTUR

Cette procédure a pour but de constituer des groupes avec les cartes qui ont trait à la mise à jour d'un même article IDS et de les conserver sur un fichier. Cette procédure produit également un listing avec les groupes et les erreurs.

Le fichier de sortie contient trois sous-ensembles constitués respectivement pour la création, la modification et la suppression.

Les groupes ainsi constitués contiennent chacun trois parties :

- 1) un en-tête où sont repris :
  - le code fonction
  - un switch valant "1"
  - le numéro IDS de l'article décrit (numéro associé à chaque type d'article)
  - le numéro de groupe (évoluant de 1 à n pour chaque type de mise à jour)
  - l'identifiant 1 : codification de l'article si c'est un article codé/codification de l'émetteur si c'est un article relation
  - l'identifiant 2 : zone remplie de caractères blancs si c'est un article codé/codification de l'émetteur si c'est un article relation
- 2) L'article proprement dit où sont reprises toutes les zones décrites "données courantes" dans la banque de données.
- 3) Une zone de complétage remplie de caractères blancs pour aligner tous les articles sur la longueur du plus grand.

./...



Une carte est considérée comme faisant partie d'un groupe si :

- son code fonction est correct
- son code carte est correct
- le couple ou triplet code carte, générique émetteur et générique récepteur est repris au répertoire des couples ou triplets possibles

Après avoir constitué les groupes, la procédure LECTUR fait appel, pour chaque type d'articles différent, à un module de test de codification.

Le module de test de codification vérifie les quatorze positions de l'identifiant de l'article.

La codification de chaque identifiant est structurée de manière hiérarchique suivant le dessin de la banque de données. Dans certains cas, la codification est constituée d'une suite de codes élémentaires.

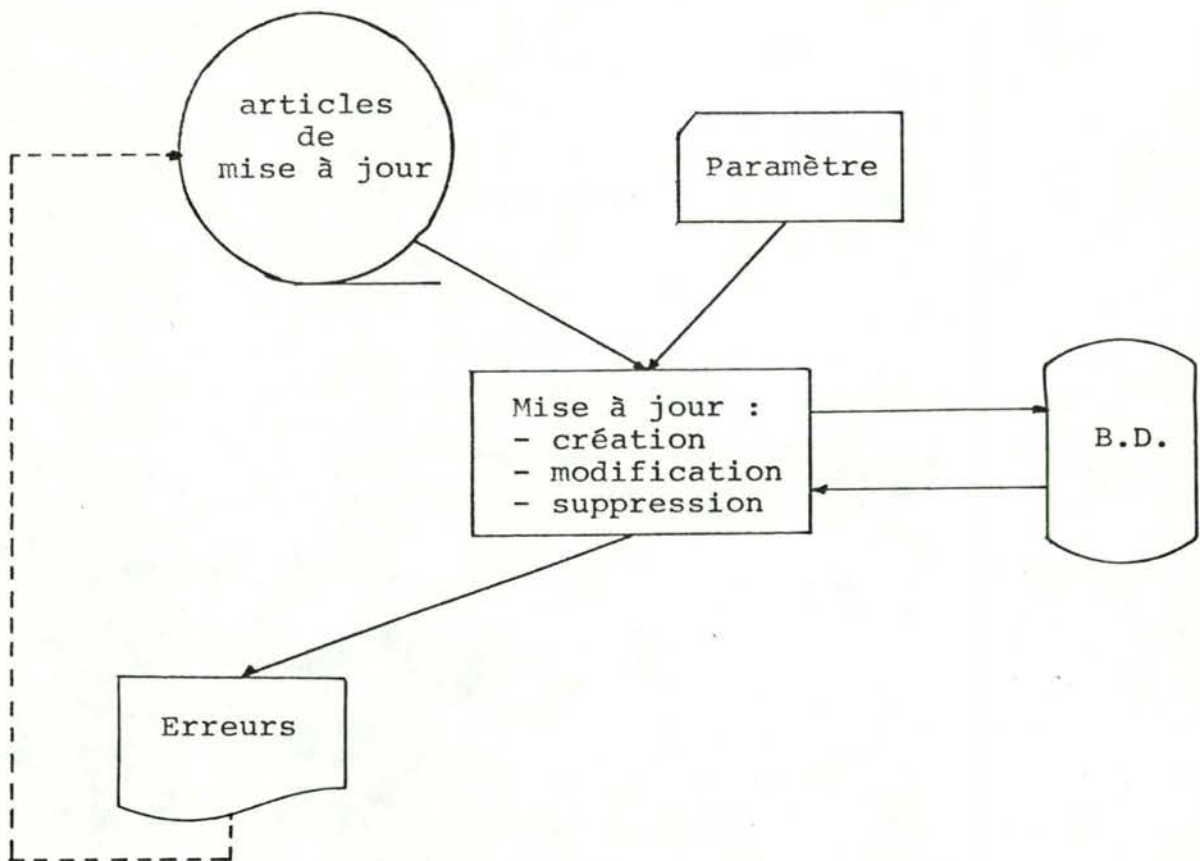
Par exemple, une section de production (S-PR) est située hiérarchiquement sous un bloc, un département et une usine. Ainsi, sa codification est composée d'un numéro d'usine, un numéro de département et un numéro de bloc.



Lorsque la codification d'un article est constituée de plusieurs codes élémentaires, le module de test de codification correspondant fait appel à un ou plusieurs autres modules. Ces modules effectuent les vérifications nécessaires sur les codes élémentaires. Il peut s'agir de tests de présence ou de contrôle de numéricité.

Par exemple, le module qui teste la codification d'une section de production fait appel au module de test de la codification du bloc, qui lui appelle le module de test de la codification du département et ainsi de suite.

Graphe de la procédure de mise à jour de la banque de données



### 3. Procédure de création CREATI

Le but de cette procédure est de charger en un ou plusieurs passages la banque de données.

L'entrée de cette procédure est le fichier constitué par la procédure de concaténation LECTUR

La procédure se décompose en :

- un programme principal
- 54 sous-programmes correspondant à chacun des 54 articles de la base de données

Principe de fonctionnement de la création :

- le programme principal lit un article
- il vérifie si le numéro de groupe de cet article est compris entre deux bornes mini-maxi (ces quatre valeurs sont données au programme par une carte paramètre)
- le programme principal appelle le sous-programme correspondant au numéro IDS de l'article
- après avoir vérifié la cohérence de l'article, le sous-programme l'insère dans la banque de données

#### - Description du programme principal

PAS 1. Ouverture du fichier d'entrée et de la banque de données  
Initialisation du programme d'impression d'erreur

- 1) si c'est une première création (header absent)  
aller à 3)
- 2) sinon garnir la date de dernière mise à jour du header avec la date du jour  
aller à 3)
- 3) lire les paramètres : numéros de groupe valides  
années valides

PAS 2. Lire un article du fichier d'entrée  
(en EOF, aller à PAS 4.)

- 1) si son code fonction  $\neq$  3 (création), aller à PAS 2.
- 2) sinon :
  - a. si son n° groupe n'est pas compris entre les bornes, aller à PAS 2.
  - b. sinon préparer la ligne erreur,  
aller à PAS 3.

PAS 3. Appeler le sous-programme correspondant au numéro IDS  
aller à PAS 2

PAS 4. Imprimer le rapport final d'activité (nombre total d'articles traités en création)

./...



PAS 5. Clôture du fichier d'entrée  
 Clôture de la banque de données  
 Terminaison du programme d'impression d'erreur

- Description générale d'un sous-programme de création d'un article

PAS 1. Rechercher le ou les maîtres de l'article à créer

PAS 2. Pour les articles qui possèdent des zones de page-range

- 1) si ces zones sont numériques, aller à PAS 3.
- 2) imprimer message d'erreur  
 aller à pas 10

PAS 3. Initialiser à zéro l'article à créer

PAS 4. Transférer toutes les zones de l'article d'entrée vers l'article de la banque de données

PAS 5. Effectuer le test de cohérence (appel d'un sous-programme)

PAS 6. Créer l'article par l'ordre IDS "STORE"

PAS 7. Fin de module

4. Procédure de modification MODIFI

Le but de cette procédure est de mettre à jour les articles existant déjà dans la base de données.

L'entrée de cette procédure est le fichier constitué par la procédure de concaténation LECTUR.

La procédure se décompose en :

- un programme principal
- 54 sous-programmes correspondant à chacun des 54 articles de la base

Principe de fonctionnement général de la modification :

- le programme principal lit un article
- il vérifie si le numéro de groupe est compris entre deux bornes mini-maxi.  
 De plus, si l'article comporte des dates, elles doivent être comprises entre deux dates mini-maxi  
 (ces quatre valeurs sont données au programme principal par une carte paramètre)
- le programme principal appelle le sous-programme correspondant au n° IDS de l'article

./...





## 5. Procédure de suppression SUPRES

Parce que la banque de données est créée par l'IDS, il est essentiel de se rappeler que la suppression d'un article entraîne automatiquement la suppression des détails de cet article et des détails de ces détails.

Le but de cette procédure est de supprimer les articles de la base de données.

L'entrée de cette procédure est le fichier constitué par la procédure de concaténation LECTUR.

Principe de fonctionnement de cette procédure :

- le programme de suppression lit un article
- il vérifie si le numéro de groupe est compris entre deux bornes mini-maxi  
(ces deux valeurs sont données au programme par une carte paramètre)
- il existe six types de séquence de traitement qui couvrent tous les cas possibles de suppression :
  - une séquence (S1) pour supprimer le header
  - une séquence (S2) pour supprimer les articles qu'on retrouve par IRTRI
  - une séquence (S3) pour supprimer les articles relations standards
  - une séquence (S4) pour supprimer les articles relations autres que les standards
  - une séquence (S5) pour supprimer les articles DPTU et SDOU
  - une séquence (S6) pour supprimer l'article ELCO
- Ce programme imprime des messages d'erreur
- Description de la procédure de suppression

PAS 1. Ouverture du fichier d'entrée et de la banque de données  
Initialisation du programme d'impression d'erreurs  
Garnir la date de dernière mise à jour du header avec la date du jour  
Lire les paramètres : numéros de groupe valides

PAS 2. Lire un article du fichier d'entrée  
(en EOF, aller en PAS 4.)

- 1) si son code fonction  $\neq$  1 (suppression)  
aller à PAS 2.
- 2) sinon :
  - a. si son numéro de groupe n'est pas compris entre les bornes, aller à PAS 2.
  - b. sinon préparer la ligne d'erreur,  
aller à PAS 3.

PAS 3. Effectuer la séquence de traitement correspondant à l'article (une des séquences S1 à S6), aller à PAS 2.

./...

PAS 4. Imprimer le rapport final d'activité (nombre total d'articles traités en suppression)

PAS 5. Clôture du fichier d'entrée  
Clôture de la banque de données  
Terminaison du programme d'impression d'erreurs

- Description des différentes séquences de traitement

- + séquence S1 pour le header
  - imprimer le message d'erreur "interdit de supprimer le header"
  - aller à PAS 4.
- + séquence S2 valable pour les articles USIN, DOVE, SOCO, TYFP, DEPT, SDVE, MARC, SDCR, FPTO, BLOC, M-CL, ENTX, DIRG, SKFP, S-AD, S-AU, S-PR, FLPR, SKPR, SKCH, S-VE, PMCL, S-DG, FPUS, CDEP, ECHX, FLDC, FPVE, FPDG, CSPE
  - rechercher l'article par un ordre IDS "IRTRI"
  - le supprimer
  - aller à PAS 2.
- + Séquence S3 valable pour les articles standards
  - rechercher l'article par un ordre IDS "IRTISM"
  - le supprimer si c'est permis
  - aller à PAS 2.
- + Séquence S4 valable pour les articles relations
  - rechercher l'article par un ordre IDS "IRTRIS"
  - le supprimer
  - aller à PAS 2.
- + Séquence S5 valable pour les articles SDOU, DPTU
  - imprimer le message "interdit de supprimer ces articles"
  - la suppression de ces articles est faite lors de la suppression des articles SDVE et DEPT associés
  - aller à PAS 2.
- + Séquence S6 valable pour l'article ELCO
  - rechercher l'article par un ordre IDS "IRTRIS"
  - le supprimer
  - aller à PAS 2.

./...



#### 4. CRITIQUE DE L'EXISTANT

=====

Le système actuel de mise à jour de la base de données est assez lourd à cause de plusieurs raisons :

- La description complète d'un article nécessite plusieurs cartes de types différents. Pour certains articles, six cartes sont nécessaires. Chaque carte contient, de ce fait, une zone variable "code carte" et une zone variable "type de carte" en plus du code fonction, du générique et de la codification.
- Le traitement de mise à jour proprement dit est précédé de deux phases destinées uniquement au traitement des cartes :
  - une phase de tri des cartes sur le code fonction, le code carte, le générique, la codification et le code séquence
  - une phase de groupage des cartes par article, sur une bande. Sur ce support intermédiaire, les articles sont de longueur fixe ; chaque article occupe la place du plus grand enregistrement possible.
- Le traitement de mise à jour s'effectue fonction par fonction : suppression, modification puis création.  
 En début de chaque type de traitement de mise à jour, une initialisation est réalisée, comprenant :
  - l'ouverture du fichier bande
  - l'ouverture de la banque de données IDS
  - l'initialisation d'un fichier d'impression des erreurs
  - la recherche de l'article de tête de la banque de données, afin de s'assurer que cette dernière existe et que le modèle n'est pas verrouillé (le modèle est verrouillé lorsqu'une simulation est en cours), et afin de mettre à jour la date de dernière mise à jour avec la date du jour
- Des contrôles sont réalisés dans chaque traitement. Cependant, l'utilisateur ne prend connaissance de ses erreurs qu'après l'enchaînement total ; il ne peut intervenir pour corriger. C'est pourquoi des valeurs par défaut sont générées, dans la mesure du possible, par les programmes d'application afin de pouvoir continuer le traitement.

Le système que nous proposons va tenter d'apporter une amélioration sur ces différents points.

./...



## 5. ETUDE DU SYSTEME PREVU DE MISE A JOUR DE LA BANQUE DE DONNEES

=====

Le but essentiel de cette nouvelle application est de permettre à l'utilisateur de mettre à jour la banque de données par l'intermédiaire d'un vidéo. Pour soutenir le dialogue entre l'utilisateur et l'ordinateur, nous avons élaboré un gestionnaire de grilles, constitué de différentes routines qui peuvent être appelées par les programmes d'application. Ainsi, les programmes d'application peuvent envoyer sur l'écran de l'utilisateur, la grille correspondant au type d'article à mettre à jour, recueillir et vérifier les données introduites. L'utilisateur travaille article par article, corrigeant ses erreurs dès qu'elles lui sont signalées à l'écran.

## 5.1 Le gestionnaire de grilles

-----

### 5.1.1 Généralités

Le gestionnaire de grilles est un système permettant de traiter des données introduites par écran et de contrôler leur validité. Il transmet ces données à un programme d'application, en exécution, qui s'adresse à ce terminal en vue d'obtenir des informations (de la même manière que le programme peut consulter un fichier). Afin de pouvoir établir cette communication entre le terminal et le programme d'application, ce dernier doit s'exécuter sous le système DAC (Direct Access). Un programme résident (le programme MAIN) assure, parmi ses fonctions, le lancement du programme d'application sous le système DAC. Afin d'assurer un maximum de souplesse à la communication, plusieurs routines ont été mises à la disposition des programmes d'application.

C'est l'ensemble du programme MAIN et des différentes routines que nous désignons par l'appellation "gestionnaire de grilles".

### 5.1.2 Le système DAC

Le système DAC (Direct Access) est un mode d'opération interactif dans lequel l'utilisateur sur terminal est en communication directe avec un programme en exécution dans le système central.

Cette communication dans les deux sens se réalise par une séquence de questions-réponses suivant le déroulement du programme. Cet échange se poursuit jusqu'à ce qu'il soit interrompu soit par le terminal, soit par le programme DAC.

Pour soutenir cette communication, le système DAC utilise GEROUT, système d'entrée-sortie à distance, qui est en fait une partie du système d'exploitation GCOS. Chaque demande d'entrée-sortie à distance se traduit par un appel à une routine du système GEROUT c'est-à-dire par une instruction MME GEROUT (Master Mode Entry). Le système GEROUT permet au programme utilisateur de communiquer avec un terminal, en réalisant les opérations suivantes :

- envoi d'un message à un terminal
- réception d'un message venant d'un terminal
- identification du terminal
- recherche du type du terminal
- déconnexion d'un terminal
- recherche du statut de la ligne
- aiguillage en accès direct

Les arguments d'une routine GEROUT sont les suivants :

- un pointeur vers une table de données (Record Pointer)
- le type de GEROUT (Operation Code)
- l'identification du terminal (Terminal ID)
- un pointeur vers le mot d'état du terminal (Status Word Pointer)

./...



### Identification du terminal (GEROUT 05)

Lorsque le programme utilisateur est prêt à communiquer avec un terminal qui a demandé à être connecté en accès direct avec ce programme, il doit donner, pour s'identifier, son nom à la routine GEROUT 05. Cette routine tient la liste de tous les terminaux qui ont fait une demande d'accès direct et parcourt cette liste afin d'identifier le terminal concerné par ce programme. Lorsqu'elle a trouvé, elle charge le "Terminal ID" et le programme utilisateur peut alors entrer en conversation avec ce terminal.

L'"Operation Code" vaut 05.

### Recherche du type du terminal (GEROUT 06)

Une fois connecté à un terminal, le programme utilisateur peut déterminer le type de ce terminal en utilisant la GEROUT 06. Le type indique si le terminal est un ordinateur, un terminal à papier ou un terminal à écran.

### Déconnexion d'un terminal (GEROUT 17)

Un programme d'application peut demander à cette routine de déconnecter un terminal en donnant son "Terminal ID".

L'"Operation Code" est égal à 17.

### Recherche du statut de la ligne (GEROUT 20)

Un programme d'application peut demander l'état de la ligne d'un terminal particulier. Cette routine fournit aussi le type de terminal.

L'"Operation Code" vaut 20.

### Aiguillage en accès direct (GEROUT 27)

Cette routine est utilisée pour passer le contrôle d'une ligne en accès direct vers un autre programme.

L'"Operation Code" est égal à 27.

./...

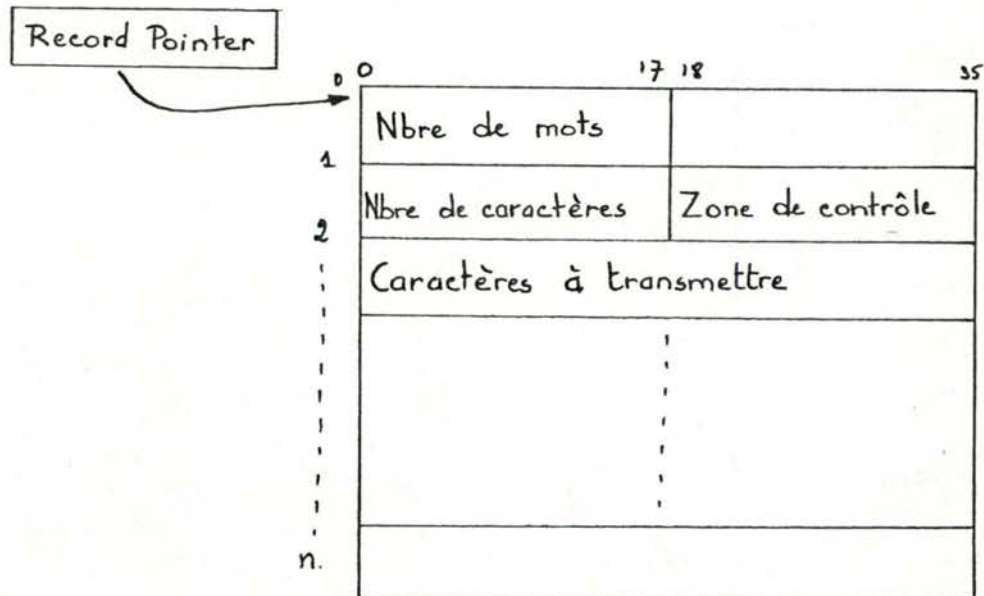
Ces différentes opérations sont décrites dans les paragraphes suivants.

Envoi d'un message à un terminal en accès direct  
(GEROUT 03)

Cette routine permet à un programme utilisateur en exécution d'envoyer un message à un ou plusieurs terminaux.

Valeurs prises par les arguments :

- le "record pointer" contient l'adresse du bloc de données. Dans le bloc de données, on trouve la longueur de ce bloc en mots, le nombre de caractères à envoyer au terminal, des indicateurs pour la routine et les caractères à transmettre.



- l'"operation code" est égal à 03
- le "Terminal ID" est un code en deux positions identifiant le terminal
- le "Status Word Pointer" contient l'adresse d'un mot où est mémorisé l'état de l'opération en cours

Les états possibles sont :

1. le système central ne peut pas prendre en considération une demande d'entrée-sortie parce que le terminal est occupé avec une entrée-sortie
2. l'envoi du message est achevé

./...

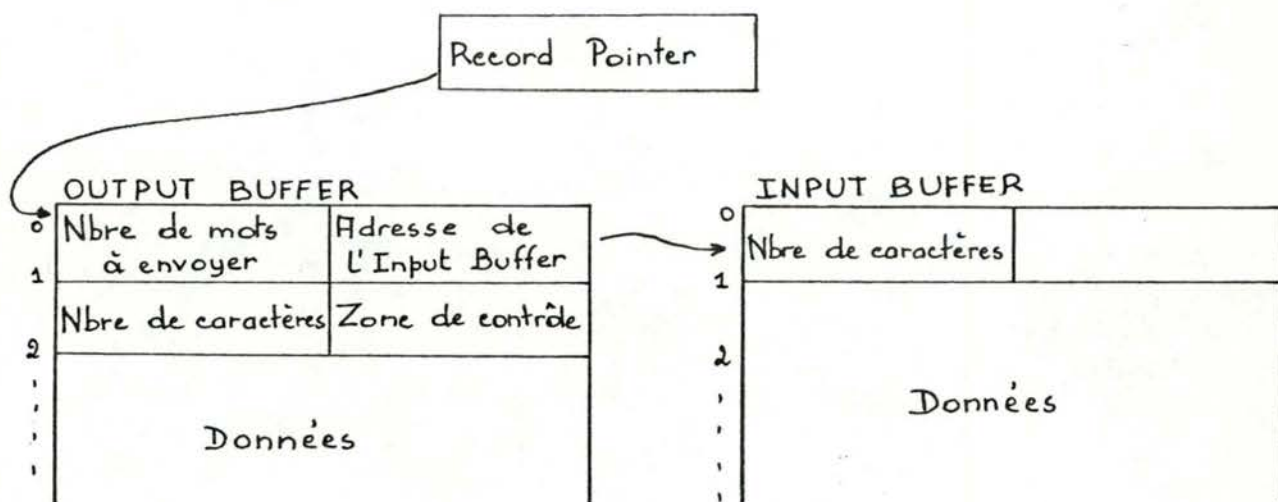
3. le terminal est connecté mais n'a rien demandé ou n'est pas connecté
4. l'utilisateur a envoyé le message qui termine l'opération (BREAK).

Envoi d'un message à un terminal puis réception d'un message (GEROUT 04)

Cette routine permet à un programme utilisateur s'exécutant dans le système central de transmettre un message à un terminal et de recevoir une réponse de celui-ci.

Valeurs prises par les arguments :

- le "record pointer" contient l'adresse du bloc de données. On trouve dans ce bloc de données, en plus des renseignements cités dans la GEROUT 03, l'adresse d'une zone destinée à recevoir le message (Input Buffer)



- l'"operation Code" est égal à 04
  - le "Terminal ID" est un code en deux positions identifiant le terminal
  - le "Status Word Pointer" contient l'adresse d'un mot où est mémorisé l'état de l'opération en cours.
- Les différents états sont les mêmes que pour la GEROUT 03 à ceci près que le terminal se trouve dans l'état 2 si l'envoi du message est complètement achevé et que la réponse a été placée dans le buffer d'entrée.



### 5.1.3 Programme MAIN

Le programme MAIN, permanent en mémoire centrale, permet à quatre utilisateurs de lancer simultanément leur propre application.

Le main examine en permanence le contenu d'une zone qui reçoit l'identification du terminal (appelée Terminal ID ou encore Station Code) de l'utilisateur dès que celui-ci s'est connecté au système. Diverses étapes se succèdent alors :

- a. envoi à l'utilisateur d'une première grille dans laquelle il lui est demandé d'introduire obligatoirement son identification (USER-ID)  
son mot de passe  
facultativement son application
- b. dès que les réponses sont introduites, le main contrôle leur validité et envoie un message d'erreur suivant le type d'anomalie rencontré. Trois tentatives sont permises avant la déconnexion programmée
- c. si le troisième paramètre est erroné ou omis, le main envoie une seconde grille affichant les applications propres à l'utilisateur. Il s'agit d'un menu dans lequel l'utilisateur doit choisir l'application sur laquelle il veut travailler.
- d. ce menu peut être affiché trois fois avant une déconnexion programmée
- e. l'opérateur reçoit ensuite la première grille de son application et peut commencer l'introduction de ses informations
- f. lorsque les données sont transmises au système, le main passe la main au programme d'application et réalise le transfert des données
- g. le main, ainsi libéré de son utilisateur peut attendre ou traiter une nouvelle connexion.

./...

Remarque : la notion de temps prend toute sa signification car pendant le traitement de ces étapes (a → g), le main examine s'il y a connexion, et lance les étapes ad hoc pour un ou plusieurs autres utilisateurs.

La procédure MAIN est activée lorsqu'un terminal demande à se connecter au système HELLO. Ce n'est donc pas le processeur frontal qui demande régulièrement aux terminaux s'ils veulent se connecter. Afin de mieux comprendre ces différentes étapes, nous présentons ici un algorithme plus détaillé.

### Algorithme du programme MAIN

#### a. Description des données et des tables

Le programme MAIN utilise une table (TTAB) de quatre éléments ; en effet quatre utilisateurs peuvent entamer simultanément une procédure de connexion. Dans chaque élément figurent une partie fixe et une partie variable.

La partie variable comprend les éléments suivants :

##### - le station code

cette variable prend la valeur de l'identification du terminal et est chargée par la routine GEROUT 05 d'identification du terminal. Elle est initialisée à zéro.

##### - adresse de l'input buffer

chargée en début de programme

##### - statut

état d'avancement dans le traitement du programme utilisateur

valeurs possibles :

- 0 initialisation
- 1 après GEROUT 05
- 2 après avoir mis le station code dans TTAB
- 3 si la USER-ID introduite par l'utilisateur est correcte
- 4 si le mot de passe introduit par l'utilisateur est correct
- 5 après avoir envoyé le menu à l'utilisateur
- 6 si l'application introduite par l'utilisateur est correcte

./...

- sous-statut
  - sert à compter le nombre d'essais lors de l'introduction de la USER-ID, du mot de passe et de l'application
  - sert à compter le temps pendant lequel le programme utilisateur attend les ressources indispensables à son chargement
  - initialisé à zéro
- sequence number (SNUMB)  
le sequence number est un numéro attribué au programme utilisateur, par le MAIN, lors de son lancement
- statut de la routine GEROUT  
il s'agit du status word, un des arguments de la GEROUT qui indique l'état de l'opération en cours. Il est initialisé à zéro.

La partie fixe comprend les instructions d'appel à une routine GEROUT. Pour appeler une routine GEROUT, il suffit donc de modifier un ou plusieurs éléments de la table, par exemple, le type de GEROUT.

Voici les instructions reprises dans la table :

MME	GEROUT	appel à la GEROUT
VFD	18/RECORD POINTER,6/OP	Réservation d'un mot (de 36 bits) dont les dix-huit premiers bits contiennent l'adresse d'un enregistrement et les six bits suivants, le type de GEROUT
ZERO	*-3	Réservation d'un mot dont la première moitié contient le statut de la GEROUT
TRA	0,1	Return



0	Station Code	Adresse de l'Input Buffer	} Partie variable
1	Statut	Sous - statut	
2	Snumb		
3	Statut de la GEROUT		
4	MME GEROUT		} Partie fixe
5	VFD 18/RECORD POINTER, 6/OP		
6	ZERO *-3		
7	TRA 0,1		

Une deuxième table (TUPA) est utilisée par le MAIN : la table des USER-ID, mot de passe, application. Cette table a quatre entrées. Chaque élément est initialisé à zéro et comprend les variables suivantes :

- USER-ID
- mot de passe
- application
- numéro de la première grille de l'application.

Le programme MAIN fait appel aux routines du système GEROUT (MME GEROUT xx) et aux routines du gestionnaire de grilles (CALL xxxxxx)

./...

b. AlgorithmeTraitement de connexion des terminaux

- GRI001 Voir dans TTAB si tous les statuts sont différents de zéro.  
 Si tous les statuts sont différents de zéro : quatre utilisateurs se connectent ; aller à GRI100, sinon possibilité d'une ou plusieurs connexions.  
     S'il y a un statut à 1 dans TTAB : la routine GEROUT 05 a été exécutée pour ce terminal ; aller à GRI010  
     sinon - si un statut vaut zéro dans TTAB  
         . exécuter la GEROUT 05 d'identification du terminal  
         . mettre statut à 1 dans TTAB  
         . aller à GRI010  
     - sinon incompatibilité avec le test en GRI001 ; déconnexion programmée ; aller à GRI120
- GRI010 Tester le station code  
 si le station code vaut zéro (le terminal ne s'est pas connecté), aller en GRI050  
 sinon aller en GRI012
- GRI050 s'il y a un station code dans TTAB, aller en GRI100  
 sinon - attendre qu'un terminal demande à se connecter (appel à une routine d'attente de GCOS : GEWAKE)  
     - aller à GRI001
- GRI012 - mettre le station code dans la VFD de la routine de recherche du type de terminal (GEROUT 20)  
 - mettre à zéro le station code dans la VFD de la GEROUT 05  
 - aller à GRI014
- GRI014 - GEROUT 20 : recherche du type de terminal  
 - tester le type de terminal  
     si type incorrect, déconnexion programmée  
         aller à GRI120  
     sinon - mémoriser le station code dans TTAB  
         - mettre le statut à 2 dans TTAB  
         - lire la grille # 1 (USER-ID, mot de passe, application) dans l'output buffer (CALL LECGRI).  
         Si la lecture n'est pas OK, déconnexion programmée ; aller à GRI120.

./...

- charger l'adresse de l'input buffer dans l'output buffer
- mettre à zéro le statut de la GEROUT 04 dans TTAB
- tester si le terminal s'est déconnecté c'est-à-dire si le station code vaut zéro  
si oui, déconnexion programmée ;  
aller à GRI120
- mettre station code de TTAB dans VFD de la GEROUT 04
- exécuter la GEROUT 04
- aller à GRI001

#### Analyse du statut du terminal de chaque élément de TTAB

---

GRI100 Prendre le premier élément de TTAB

GRI110 Tester le statut du terminal par rapport à 1  
si statut du terminal  $\leq 1$ , aller à GRI150  
sinon tester le statut de la GEROUT 04 dans TTAB

- si statut GEROUT 04 vaut zéro, aller à GRI150
- sinon - remise du statut de la GEROUT 04 à zéro
  - test de déconnexion
    - . si le terminal s'est déconnecté, traitement de déconnexion ; aller à GRI120
    - . sinon test du statut
      - si le statut vaut 0, fin
      - si le statut vaut 1, déconnexion programmée, aller à GRI120
      - si le statut vaut 2, aller à GRI200
      - si le statut vaut 3, aller à GRI500
      - si le statut vaut 4, aller à GRI600
      - si le statut vaut 5, aller à GRI700
      - si le statut vaut 6, aller à GRI400

GRI150 Tester le station code dans la VFD de la GEROUT 05  
Si le station code est différent de zéro, aller à GRI012

./...



Sinon - passer à l'élément suivant dans TTAB  
 - si on est en fin de table, aller à  
 GRI160  
 sinon aller en GRI155

GRI160 revenir au premier élément  
 aller à GRI155

GRI155 Tester si on a parcouru toute la table  
 Si oui - attendre qu'un terminal demande à se  
 connecter (appel à la routine GEWAKE)  
 - aller à GRI001  
 Sinon aller à GRI110

Statut = 2 : analyse du USER-ID introduit dans la  
 première grille

GRI200 -Test de validité de l'identification de l'uti-  
 lisateur (USER-ID)  
 Si USER-ID valide, mettre le statut (dans TTAB)  
 à 3  
 aller à GRI205  
 sinon aller à GRI215

GRI215 ajouter 1 au sous-statut de TTAB  
 tester le sous-statut par rapport à 3  
 si le sous-statut est  $\geq 3$ , on a au moins fait  
 trois essais dans l'introduction de la  
 USER-ID, déconnexion programmée ; aller  
 à GRI120  
 sinon - envoi du message à l'utilisateur :  
 "USER-ID invalide" (CALL INSTXT)  
 - aller à GRI214

GRI214 - exécuter la GEROUT 04 ;  
 - aller à GRI120

GRI120 - Passer à l'élément suivant dans TTAB  
 - aller à GRI001

Statut = 3 : analyse du mot de passe introduit dans  
 la première grille

GRI500 - Test de validité du mot de passe  
 si le mot de passe est valide, mettre le  
 statut (dans TTAB) à 4 ; aller à GRI210 ;  
 sinon aller à GRI220

GRI220 - ajouter 1 au sous-statut de TTAB  
 - tester le sous-statut par rapport à 3  
 si le sous-statut est  $\geq 3$ , on a au moins  
 fait trois essais dans l'introduction du  
 mot de passe ; déconnexion programmée ;

./...

aller à GRI120 ;  
 sinon - envoi du message à l'utilisateur :  
       "Mot de passe invalide".  
       (CALL INSTXT) ;  
       - aller à GRI214 ;

Statut = 4 : analyse de l'application introduite dans la première grille

- GRI600 - Test de validité de l'application  
           si l'application est valide, aller à GRI300  
           sinon aller à GRI225
- GRI225 - Lire la grille #2 du MENU (CALL LECGRI) dans  
           l'output buffer ;  
           si la lecture n'est pas OK, déconnexion pro-  
           grammée ;  
                                                           aller à GRI120 ;  
           sinon - constitution du menu ;  
                   - envoi du menu à l'utilisateur (CALL  
                   INSTXT) ;  
                   - mettre le statut (dans TTAB) à 5 ;  
                   - aller à GRI214 ;
- GRI300 - Attribuer un sequence number (SNUMB) à  
           l'application et le mettre dans TTAB ;  
           - lancer l'application  
           - lire la première grille de l'application  
           (CALL LECGRI) dans l'output buffer  
           si la lecture n'est pas OK, déconnexion  
           programmée  
                                                           aller à GRI120  
           sinon tester zone de contrôle (dans le deu-  
           xième mot de l'output buffer)  
           si zone = 0, aller à GRI250 ;  
           si zone = -1, aller à GRI270 ;  
           si zone = 1 : dans ce cas, il s'agit de  
           lancer une application dont la première  
           grille doit être complétée par l'utili-  
           sateur ;  
           mettre le statut (dans TTAB) à 6 ;  
           aller à GRI214 ;
- GRI250 dans ce cas, il s'agit de lancer une applica-  
           tion qui s'exécute sans affichage de grilles  
           - envoyer à l'utilisateur le sequence number  
           (SNUMB) de l'application (CALL INSTXT, exé-  
           cuter la GEROUT 03 puis appel à la routine  
           GEWAKE) ;  
           - déconnexion programmée ;  
           - aller à GRI120 ;

./...



- GRI270 dans ce cas, il s'agit de lancer une application dont la première grille visualise des valeurs sans demander de réponse.
- envoyer à l'utilisateur la grille formatée avec les libellés (exécuter la GEROUT 03) ;
  - exécuter la GEROUT 27 qui met en communication le programme d'application avec le terminal
- c'est le programme d'application qui chargera les variables à partir des fichiers ;
- réinitialiser l'élément de TTAB correspondant à ce terminal ;
  - aller à GRI120 ;

Statut = 5 : analyse de l'application introduite dans le cas où le menu a été visualisé une fois

- GRI700 Test de validité de l'application
- si l'application est valide, aller à GRI300 ;
- sinon test du numéro de grille (dans TUPA) :
- si numéro de grille = 0 (on n'a pas encore lu la première grille de l'application)
  - aller à GRI730 ;
  - sinon aller à GRI710 ;

- GRI730
- constitution du menu ;
  - envoi du menu à l'utilisateur (CALL INSTXT) ;
  - aller à GRI214 ;

- GRI710
- ajouter 1 au sous-statut de TTAB ;
  - tester le sous-statut par rapport à 3
  - si le sous-statut est  $\geq 3$ , on a au moins fait trois essais dans l'introduction de l'application ; déconnexion programmée ;
  - aller à GRI120 ;
  - sinon - constitution du menu ;
  - envoi du menu à l'utilisateur ;
  - aller à GRI214 ;

Statut = 6 : l'application est correcte, passer la main au programme utilisateur

- GRI400
- exécuter la GEROUT 27 qui met en communication le programme d'application et le terminal ;
  - réaliser le transfert des données vers le programme d'application. C'est le MAIN qui a envoyé à l'utilisateur la première grille de son programme.
- Le main a reçu les données introduites par l'utilisateur (GEROUT 04) ; il doit maintenant les transférer au programme utilisateur qui vient d'être lancé et qui doit traiter ces données ;

./...



- test de fin de transfert  
Pour que ce transfert se déroule sans problèmes, il faut que le programme utilisateur soit en mémoire et qu'il ait acquis les ressources nécessaires. Cette allocation de la mémoire et des ressources peut demander un certain temps.  
si le transfert est terminé,
  - réinitialiser l'élément de TTAB correspondant à ce terminal ;
  - aller à GRI120 ;
- sinon - ajouter 1 au sous-statut de TTAB
  - tester le sous-statut par rapport à 10
    - si le sous-statut est  $\geq 10$ ,
      - envoyer un message à l'utilisateur,
      - déconnexion programmée ;
      - aller à GRI120
    - sinon - garnir le statut de la GEROUT dans TTAB ;
      - aller à GRI120 ;

./...

#### 5.1.4 Routines du gestionnaire de grilles

Les routines du gestionnaire de grilles, mises à la disposition des programmes utilisateurs, remplissent les fonctions suivantes :

- aiguillage du contrôle de la ligne
- connexion du terminal
- transformation d'une grille à zones variables en une grille à zones fixes
- lecture d'une grille avec ses formats et ses fourchettes
- lecture d'une grille sans formats ni fourchettes
- insertion de texte dans une grille
- affichage d'une grille et réception des zones variables
- affichage d'une grille
- package
- dépackage
- attente
- déconnexion du terminal

# 1. Routine d'aiguillage du contrôle de la ligne (LINSW)

## - But

Le but de cette routine est de passer le contrôle de la ligne d'une application à une autre. Le programme à qui le contrôle est donné, doit débiter par un appel à la routine INQTER afin de recevoir l'identification du terminal (station code).

## - Instruction d'appel et arguments

CALL LINSW (ITERM, NAME, STAT)

ITERM - station code du terminal

- variable positionnée par la routine INQTER

NAME - nom de l'application qui reçoit le contrôle de la ligne

STAT - statut de la routine

- positionné par la routine

- valeurs possible

0 : déroulement correct

1 : déroulement incorrect

## - algorithme

PAS 1 - initialisation du statut à zéro ;

PAS 2 - tester le NAME

si NAME = 0 mettre le statut à 1 ;  
aller à PAS 5 ;

sinon aller à PAS 3 ;

PAS 3 - garnir la VFD de la GEROUT 27 avec le station code ;

- exécuter la GEROUT 27 ;

PAS 4 - tester si la GEROUT 27 s'est bien déroulée (tester le Status Word de la GEROUT)

si oui : aller à PAS 5 ;

sinon : mettre le statut à 1 ;

aller à PAS 5 ;

PAS 5 - fin.

./...



## 2. Routine de connexion (INQTER)

### - But

Le but de la routine INQTER est de déterminer si quelqu'un s'est connecté et d'identifier le terminal.

### - Instruction d'appel et arguments

CALL INQTER (ITERM, NAME)

ITERM - station code du terminal  
 - positionné par la routine  
 - valeurs possibles

0 : pas de terminal connecté  
 station code : terminal connecté

NAME - nom de l'application qui a le contrôle du terminal, sauf s'il s'agit de la première application  
 - valeurs possibles

0 : s'il s'agit de la première application, c'est-à-dire une application qui n'a pas reçu le contrôle par un LINSW  
 nom sous lequel on a fait un LINSW dans l'application précédente

### - algorithme

PAS 1 - initialisation du station code à zéro ;

PAS 2 - exécuter la GEROUT 05 ;

PAS 3 - Tester si la GEROUT 05 a positionné le station code du terminal

- si oui - mettre le station code dans ITERM ;

- mettre le station code dans la VFD de la GEROUT 20 ;

- exécuter la GEROUT 20 ;

- aller à PAS 4 ;

- sinon : aller à PAS 5 ;

PAS 4 - Tester le type du terminal, donné par la GEROUT 20

- si le type du terminal est correct :  
 aller à PAS 5 ;

- sinon - garnir la VFD de la GEROUT 17 avec le station code ;

- exécuter la GEROUT 17 ;

- mettre le station code à zéro ;

- aller à PAS 5 ;

PAS 5 - fin.

./...

### 3. Routine transformant une grille à zones variables en une grille à zones fixes (BLDFIX)

#### - But

Le but de cette routine est de transformer, en mémoire, les zones variables de la grille en zones fixes. La grille n'est pas modifiée sur le disque. La routine BLDFIX est nécessaire lors de l'enchaînement de deux grilles ; on rend fixes les variables de la première grille, afin que le curseur se positionne sur les variables de la deuxième grille sans parcourir celles de la première.

#### - Instruction d'appel et arguments

CALL BLDFIX (BUFOUT, BUFIN)

BUFOUT - zone qui contient la description de la grille c'est-à-dire : grille 512 mots  
format 64 mots  
fourchettes 448 mots

- la partie grille est modifiée par la routine : les zones variables sont chargées avec les valeurs et transformées en zones fixes

BUFIN - zone contenant les valeurs introduites par l'utilisateur lors du talk

#c HT HT HT ETX BUFIN

ETX : fin de message (End of Text)

510 d #c 0 ESC N FFD DEL ESC ESC2 RPRM LIGNE COLONNE FS zone fixe

GS type de zone variable FS ESC Σ PR BUFOUT

FS : indicateur de zone fixe

GS : indicateur de zone variable

HT : fin de zone (Horizontal Tabulation).

./...

- Algorithme

- PAS 1 - Se positionner après le premier mot dans BUFIN et après les deux premiers mots dans BUFOUT.
- PAS 2 - Tester dans BUFOUT s'il s'agit d'une variable ou d'une zone fixe.  
- si c'est une zone variable, aller à PAS 3 ;  
- sinon aller à PAS 4 ;
- PAS 3 - remplacer l'indicateur de zone "GS" par un caractère de remplissage (delete character);  
- mettre un caractère de remplissage dans le caractère "type de variable" ;  
- transfert de la variable dans les positions suivantes de BUFOUT ;  
- aller à PAS 5 ;
- PAS 4 - remplacer le caractère "FS" par un caractère de remplissage (delete character) ;  
- aller à PAS 6 ;
- PAS 5 - se positionner sur la zone suivante dans BUFIN ;
- PAS 6 - se positionner sur la zone suivante dans BUFOUT ;
- PAS 7 - tester si c'est la dernière zone de BUFIN  
si oui, aller à PAS 8 ;  
sinon aller à PAS 3 ;
- PAS 8 - fin.



#### 4. Routine de lecture d'une grille avec ses formats et ses fourchettes (GRILEC)

##### - But

Le but de cette routine est de lire une grille sur disque et de l'amener en mémoire avec les formats et les fourchettes.

##### - Instruction d'appel et arguments

CALL (BUFOUT, NOGRI, STAT)

BUFOUT - zone qui reçoit la grille

. grille	512 mots
. formats	64 mots
. fourchettes	448 mots

NOGRI - numéro de la grille à aller chercher sur disque

- valeurs possibles : NOGRI > 0  
NOGRI < 0

STAT - statut de la routine

- positionné par la routine

- valeurs possibles :

0	déroulement correct
1	lecture disque incorrecte
2	numéro de grille n'existe pas sur disque
3	NOGRI est égal à zéro

#### 5. Routine de lecture d'une grille sans formats ni fourchettes (LECGRI)

##### - But

Le but de cette routine est de lire une grille sur disque et l'amener en mémoire, sans les formats ni les fourchettes.

##### - Instruction d'appel et arguments

CALL LECGRI (BUFOUT, NOGRI, STAT)

BUFOUT - zone qui reçoit la grille

. grille	512 mots
----------	----------

NOGRI - numéro de la grille à aller chercher sur disque

valeurs possibles : NOGRI > 0  
NOGRI < 0

STAT - statut de la routine

- positionné par la routine

- valeurs possibles

0	: déroulement correct
1	: lecture disque incorrecte
2	: le numéro de grille n'existe pas sur le fichier
3	: NOGRI = 0

./...

6. Routine d'insertion de texte dans une grille (INSTXT)- But

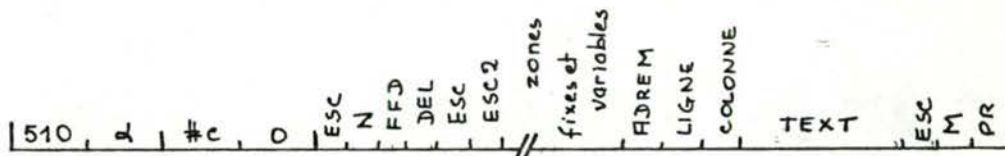
Le but de cette routine est d'insérer un texte dans BUFOUT ; ce texte sera envoyé comme zone fixe à l'écran par la routine TALK ou SEND. Aucun contrôle n'est effectué quant à l'emplacement de ce texte dans la grille.

- Instruction d'appel et arguments

CALL INSTXT (BUFOUT, LIGNE, COLON, TEXT, NBR)

BUFOUT - zone qui contient la description de la grille, c'est-à-dire : grille 512 mots  
format 64 mots  
fourchettes 448 mots

- la partie grille (qui est en ASCII) est modifiée par la routine : le texte (en BCD) est transformé en ASCII et est ajouté en tant que zone fixe, derrière les autres zones fixes et variables.



α : adresse de la zone INPUT

#C : nombre de caractères (auquel on a ajouté le nombre de caractères du texte inséré)

ADREM : caractère de contrôle qui indique une nouvelle zone

LIGNE - ligne de l'écran où est inséré le texte

COLON - colonne de l'écran où débute le texte sur la ligne

TEXT - libellé du texte à insérer

NBR - nombre de caractères du texte.

- Algorithme

PAS 1 - se positionner après les zones fixes et variables ;

PAS 2 - insérer ADREM, LIGNE, COLON ;

PAS 3 - transformer le texte en ASCII ;

./...

- PAS 4 - insérer le texte après COLON ;
- PAS 5 - mettre à jour le nombre de caractères ;
- PAS 6 - mettre les caractères de contrôle de fin de message (ESC, M et PR) ;
- PAS 7 - fin.



## 7. Routine d'affichage d'une grille et de réception des zones variables (TALK)

### - But

Le but de cette routine est d'afficher une grille à l'écran et de permettre l'introduction de données dans les zones variables.

### - Instruction d'appel et arguments

CALL TALK (ITERM, BUFOUT, BUFIN, STAT)

ITERM : station code du terminal

BUFOUT : zone qui contient la description de la grille; la partie grille est envoyée à l'écran

BUFIN : - zone garnie par l'introduction des données  
 - après chaque variable introduite, un caractère de contrôle HT est inséré dans BUFIN (quand l'utilisateur frappe la touche TAB pour passer à la zone suivante)  
 Le caractère de contrôle ETX est ajouté après la dernière variable quand l'utilisateur frappe la touche "Return"

STAT : statut de la routine

valeurs possibles : 0 si la routine s'est déroulée correctement  
 ≠ 0 sinon

### - Algorithme

- PAS 1 - Initialiser le statut de la routine à zéro ;
- PAS 2 - Mettre le station code du terminal dans la VFD de la GEROUT 04 ;  
 - Mettre l'adresse de BUFOUT dans la VFD de la GEROUT 04 ;  
 - Mettre l'adresse de BUFIN dans BUFOUT ;
- PAS 3 - Exécuter la GEROUT 04 ;  
 attendre que l'utilisateur ait entré toutes ses données
- PAS 4 - Tester le statut de la GEROUT 04  
 si statut est égal à zéro (erreur),  
 mettre une valeur (≠ 0) dans le statut de la routine  
 aller à PAS 5 ;  
 sinon aller à PAS 5 ;
- PAS 5 - Fin.

./...

## 8. Routine d'affichage d'une grille (SEND)

### - But

Le but de cette routine est d'envoyer une grille à l'écran sans permettre l'introduction de variables.

### - Instruction d'appel et arguments

CALL SEND (ITERM, BUFOUT, BUFIN, STAT)

ITERM : station code du terminal

BUFOUT : zone qui contient la description de la grille; la partie grille est envoyée à l'écran

BUFIN : argument sans utilité, si ce n'est pour garder les mêmes arguments que dans le TALK.

STAT : statut de la routine

valeurs possibles : 0 si la routine s'est déroulée correctement  
 ≠ 0 sinon

### - Algorithme

PAS 1 - initialiser le statut de la routine à zéro ;

PAS 2 - mettre le station code du terminal dans la VFD de la GEROUT 03 ;  
 - mettre l'adresse de BUFOUT dans la VFD de la GEROUT 03 ;

PAS 3 - exécuter la GEROUT 03 ;

PAS 4 - Tester le statut de la GEROUT 03  
 si statut = 0 (erreur), mettre une valeur (≠ 0) dans le statut de la routine ;  
 aller à PAS 5 ;  
 sinon aller à PAS 5 ;

PAS 5 - fin.

./...

## 9. Routine de package (PACK)

### - But

Le but de cette routine est de transformer une zone chargée dans le programme utilisateur (zone en BCD), en une zone en ASCII qui peut être envoyée à l'écran par un TALK ou un SEND.

### - Instruction d'appel et arguments

CALL PACK (BUFOUT, ECRAN, CARD)

BUFOUT - zone qui contient la description de la grille, c'est-à-dire : grille 512 mots  
format 64 mots  
fourchettes 448 mots

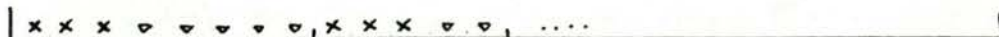
ECRAN - zone en ASCII chargée par la routine à partir de CARD et en fonction du format de BUFOUT. Des caractères de contrôle sont générés en début et en fin de la zone ainsi qu'entre les zones variables.



Le premier mot sera chargé par le TALK ou le SEND qui suivra.

ESC, M, PR, FF, ETX : caractères de contrôle  
#C : nombre de caractères de la zone à partir du troisième mot (CAR)

CARD - zone en BCD de 100 mots où chaque variable occupe la longueur décrite dans le format, la variable est éventuellement complétée par des blancs.



### - Algorithme

- PACK 1 - initialiser les compteurs ;  
- générer les quatre premiers caractères de contrôle dans ECRAN (ESC, M, PR et FF) ;  
- ajouter 4 au nombre de caractères ;

./...



- PACK 2 - prendre dans BUFOUT le format correspondant à la variable traitée ;  
 - tester le format de la variable,  
   - si type = A (alphanumérique), aller à ALPHA 1 ;  
   - si type = I (entier), aller à INT 1 ;  
   - si type = F (flottant), aller à FLOT 1 ;  
   - sinon aller à ABORT ;
- ALPHA 1 - transformation de la zone BCD en ASCII dans une zone intermédiaire ;  
 - transfert de la variable vers la zone ECRAN sur la longueur indiquée dans le format ;  
 - ajouter la longueur dans le compteur du nombre de caractères ;  
 - se positionner sur la zone suivante dans CARD ;  
 - aller à PACEND ;
- INT 1 - parcourir les caractères blancs non significatifs de gauche dans la zone variable de CARD ;
- INT 2 - analyse du caractère suivant :  
   si numérique, aller à INT 3 ;  
   sinon, aller à INT 6 ;
- INT 3 - transformer le chiffre BCD en ASCII ;  
 - transférer ce chiffre dans la zone ECRAN ;  
 - ajouter un au compteur du nombre de caractères ;
- INT 4 - se positionner sur le caractère suivant dans la zone courante de CARD ;  
 - si dernier caractère de la zone, aller à PACEND ;  
   sinon aller à INT 5 ;
- INT 5 - test du caractère :  
   si le caractère est non numérique, aller à ABORT ;  
   sinon aller à INT 3 ;
- INT 6 - si le caractère est égal à +, aller à INT 4 (le signe + n'est pas transféré) ;  
   sinon aller à INT 7 ;
- INT 7 - si le caractère est égal à - ,  
   - transformer le caractère - en ASCII ;  
   - transférer ce caractère dans la zone ECRAN ;  
   - ajouter un au compteur du nombre de caractères ;

./...

- aller en INT 4 ;  
sinon aller à INT 8 ;
- INT 8 - si le caractère est égal à \*,
  - mettre un caractère de contrôle dans la zone ECRAN, caractère qui provoquera le clignotement de la variable à l'écran ;
  - ajouter un au compteur du nombre de caractères,
  - se positionner sur le caractère suivant de la zone de CARD ;
  - si dernier caractère de la zone, aller à PACEND ;  
sinon aller à INT 9 ;
- INT 9 - test du caractère :
  - si le caractère est différent de \* ,  
aller à ABORT ;  
sinon aller à INT 8 ;
- FLOT 1 - initialiser un switch à zéro ; ce switch sera positionné à un lorsqu'on rencontrera un point ;
  - aller à FLOT 2 ;
- FLOT 2 - parcourir les caractères blancs non significatifs de gauche dans la zone variable de CARD ;
- FLOT 3 - analyse du caractère suivant :
  - si le caractère est numérique, aller à FLOT 4,
  - sinon aller à FLOT 10 ;
- FLOT 4 - transformer le chiffre BCD en ASCII ;
  - transférer ce chiffre dans la zone ECRAN ;
  - ajouter un au compteur du nombre de caractères ;
- FLOT 5 - se positionner sur le caractère suivant de la zone courante de CARD ;
  - si dernier caractère de la zone, aller à PACEND ;  
sinon aller à FLOT 6 ;
- FLOT 6 - test du caractère :
  - si le caractère est non numérique, aller à FLOT 7 ;  
sinon aller à FLOT 4 ;

./...

- FLOT 7 - si le caractère est égal à point, aller à FLOT 9 ;  
sinon aller à FLOT 8 ;
- FLOT 8 - si le caractère est égal à blanc, aller à FLOT 5 ;  
sinon aller à ABORT ;
- FLOT 9 - test du switch :  
si switch différent de zéro, aller à ABORT ;  
sinon - transformer le point en ASCII ;  
- transférer le point dans la zone ECRAN ;  
- ajouter un au compteur du nombre de caractères ;  
- mettre le switch à un ;  
- aller à FLOT 5 ;
- FLOT 10 - si le caractère est égal à +, aller à FLOT 5 ;  
sinon aller à FLOT 11 ;
- FLOT 11 - si le caractère est égal à -,  
- transformer le signe - en ASCII ;  
- transférer le signe - dans la zone ECRAN ;  
- ajouter un au compteur du nombre de caractères ;  
- aller à FLOT 5 ;  
sinon aller à FLOT 12 ;
- FLOT 12 - si le caractère est égal à point,  
- mettre le switch à un ;  
- tester le caractère suivant :  
si le caractère suivant est égal à \*, aller à FLOT 13 ;  
sinon - transformer le point ASCII ;  
- transférer le point dans la zone ECRAN ;  
- ajouter un au compteur du nombre de caractères ;  
- aller à FLOT 5 ;  
sinon aller à FLOT 13 ;
- FLOT 13 - si le caractère est égal à \*,  
- mettre un caractère de contrôle dans la zone ECRAN, caractère qui provoquera le clignotement de la variable à l'écran ;  
- ajouter un au compteur du nombre de caractères ;

./...



- se positionner sur le caractère suivant de la zone de CARD ;
- si dernier caractère de la zone, aller à PACEND ;
- sinon aller à FLOT 14 ;

FLOT 14 - test du caractère :  
 si le caractère est différent de \* ,  
 aller à ABORT ;  
 sinon aller à FLOT 13 ;

PACEND - tester, en consultant le format, si la variable que l'on vient de traiter est la dernière ;  
 si oui, aller à PACEND 1 ;  
 sinon - générer le caractère de contrôle HT (fin de variable) dans ECRAN ;  
 - ajouter un au compteur du nombre de caractères ;  
 - se positionner sur la variable suivante ;  
 - aller à PACK 2

PACEND 1 - générer dans ECRAN, les caractères ETX, PR  
 - ajouter deux au compteur du nombre de caractères ;  
 - stocker le compteur du nombre de caractères dans ECRAN ;  
 - fin ;

ABORT - fin anormale : exécuter une routine de DUMP.

10. Routine de dépackage (DEPACK)- But

Le but de cette routine est de constituer, à partir de la zone chargée lors du TALK par l'utilisateur (zone en ASCII), une zone en BCD exploitable dans le programme utilisateur. Si l'utilisateur a introduit une variable erronée (valeur en dehors des fourchettes), la routine se charge de renvoyer la grille à l'utilisateur en faisant clignoter ces zones erronées. Le cycle se poursuit tant que toutes les variables ne sont pas dans les fourchettes.

- Instruction d'appel et arguments

CALL DEPACK (BUFOUT, INPUT, OUTPUT, CARD, ITERM, STAT)

BUFOUT - zone qui contient la description de la grille

grille	512 mots
format	64 mots
fourchettes	448 mots

INPUT - zone chargée lors du TALK par l'utilisateur. Cette zone est en ASCII ; elle contient les variables introduites à l'écran, séparées par le caractère de contrôle HT.

#c		HT		HT		HT		ETX
----	--	----	--	----	--	----	--	-----

OUTPUT - #C : nombre de caractères qui suivent.  
- zone en ASCII utilisée par le DEPACK lorsqu'il détecte des erreurs de fourchettes. C'est cette zone qui est renvoyée à l'écran avec les variables erronées.

	#c	O	ESC	M	PR	FFD		HT		HT		ETX	PR
1 mot	1 mot												

Le premier mot est chargé par le TALK.

#C : nombre de caractères à partir du troisième mot.

ESC, M, PR, FFD, ETX : caractères de contrôle

Entre FFD et ETX : INPUT sans le premier mot (#C) ni ETX.

./...

- CARD - zone correcte en BCD, exploitable par le programme utilisateur. Cette zone est chargée par la routine. Chaque variable occupe la longueur décrite dans le format. Une variable est éventuellement complétée par des blancs.  
Les zones alphanumériques et flottantes sont cadrées à gauche. Les zones entières sont cadrées à droite.
- ITERM - station code du terminal  
nécessaire lors du TALK éventuel
- STAT - statut de la routine  
valeurs possibles : 0 si OK  
                  ≠ 0 sinon.



- Algorithme

- DEPACK
- initialiser les compteurs ;
  - initialiser le statut à zéro ;
  - initialiser un switch d'erreur à zéro ;
  - mettre les deux premiers mots de OUTPUT ;
  - charger le troisième mot de OUTPUT avec les caractères de contrôle ;
  - ajouter 4 au nombre de caractères (CAR) ;
  - insérer le caractère de contrôle PR dans OUTPUT (après ETX) ;
  - se positionner sur la première variable de INPUT ;
- DEPACK 1
- mettre le switch erreur à zéro ;
  - prendre dans BUFOUT le format et les fourchettes correspondant à la variable traitée ;
  - tester le format de la variable ;
    - si type = A (alphanumérique), aller à ALPHA 1 ;
    - si type = I (entier), aller à INT 1 ;
    - si type = F (flottant), aller à FLOT 1 ;
    - sinon positionner le statut à 1 et aller à FIN ;
- ALPHA 1
- tester la variable d'INPUT
    - si = blanc, aller à ALPHA 2 ;
    - sinon aller à ALPHA 3 ;
- ALPHA 2
- tester si la zone est obligatoire
    - si oui, aller à ALPHA 4 ;
    - sinon, aller à ALPHA 5 ;
- ALPHA 3
- tester si la zone est comprise entre ses fourchettes
    - si oui - transférer la zone dans OUTPUT ;
    - ajouter la longueur de la zone dans CAR ;
    - transformer la zone ASCII en BCD ;
    - transférer la zone dans CARD ;
    - générer des blancs dans CARD si la longueur de la zone d'INPUT est inférieure à la longueur du FORMAT ;
    - aller à DEPAC 10 ;
    - sinon - transférer la zone dans

./...

- OUTPUT avec un caractère BLINK dans la première position ;
  - ajouter la longueur de la zone dans CAR ;
  - mettre le switch d'erreur à 1 ;
  - aller à DEPAC 10 ;
- ALPHA 4 - générer le caractère BLINK dans OUTPUT ;
  - additionner 1 à CAR ;
  - mettre le switch d'erreur à 1 ;
  - aller à DEPAC 10 ;
- ALPHA 5 - générer des blancs sur la longueur du FORMAT dans CARD ;
  - aller à DEPAC 10 ;
- INT 1 - initialiser switch entier à 0 ;
  - tester la variable d'INPUT
    - si = blanc, aller à INT 2 ;
    - sinon aller à INT 3 ;
- INT 2 - tester si la zone est obligatoire
  - si oui, aller à INT 5 ;
  - sinon, aller à INT 6 ;
- INT 3 - tester la zone
  - INT 3.1 - parcourir les caractères blancs non significatifs de gauche
  - INT 3.2 - analyse du caractère suivant
    - si numérique, aller à INT 3.3 ;
    - sinon, aller à INT 3.5 ;
  - INT 3.3 - se positionner sur le caractère suivant dans la zone ;
    - si fin de zone, aller à INT 4 ;
    - sinon aller à INT 3.4 ;
  - INT 3.4 - si le caractère est non numérique, aller à INT 3.8,
    - sinon aller à INT 3.3 ;
  - INT 3.5 - si caractère = +, aller à INT 3.3 ;
  - INT 3.6 - si caractère = -, transformer le caractère - en BCD ;
    - aller à INT 3.3 ;
  - INT 3.7 - si caractère = blink, mettre 1 dans switch entier, aller à INT 4 ;
  - INT 3.8 - si caractère ≠ blanc, mettre switch entier à 1, aller à INT 4 ;
  - INT 3.9 - si caractère et le reste de la zone est à blanc, aller à INT 4 ; sinon mettre le switch entier à 1 et aller à INT 4 ;

./...

- INT 4 - tester le switch entier  
     si = 0 - transférer la zone dans OUTPUT ;  
             - ajouter la longueur de la zone dans CAR  
             - transformer la zone ASCII en BCD ;  
             - transférer la zone dans CARD en la cadrant à droite (générer des blancs éventuels à gauche) ;  
             - aller à DEPAC 10 ;  
     si = 1 - ajouter la longueur de la zone dans CAR ;  
             - transférer la zone dans OUTPUT avec un caractère BLINK en première position ;  
             - mettre le switch erreur à 1 ;  
             - aller à DEPAC 10 ;
- INT 5 - générer le caractère BLINK dans OUTPUT ;  
     - additionner 1 à CAR ;  
     - mettre le switch d'erreur à 1 ;  
     - aller à DEPAC 10 ;
- INT 6 - générer des blancs sur la longueur du FORMAT , dans CARD ;  
     - aller à DEPAC 10 ;
- FLOT 1 - Initialiser le switch flottant à 0 ;  
     - tester la variable d'INPUT  
         si = blanc, aller à FLOT 2 ;  
         sinon aller à FLOT 3 ;
- FLOT 2 - tester si la zone est obligatoire  
     si oui, aller à FLOT 5 ;  
     sinon aller à FLOT 6 ;
- FLOT 3 - tester la zone  
     FLOT 3.1 - initialiser un switch point à 0 ;  
     FLOT 3.2 - parcourir les caractères blancs non significatifs de gauche ;  
     FLOT 3.3 - analyse du caractère suivant  
             si le caractère est numérique, aller à FLOT 3.4 ;  
             sinon aller à FLOT 3.10 ;  
     FLOT 3.4 - se positionner sur le caractère suivant  
             - si fin de zone, aller à FLOT 4 ;  
     FLOT 3.5 - si le caractère est non numérique, aller à FLOT 3.6 ;  
             - sinon aller à FLOT 3.4 ;  
     FLOT 3.6 - si le caractère = ., aller à FLOT 3.9 ;

./...



FLOT 3.7 - si le caractère = blanc, aller à FLOT 3.8  
 sinon mettre le switch flottant à 1 ;  
 aller à FLOT 4 ;

FLOT 3.8 - si le reste de la zone = blanc, aller à FLOT 4 ;  
 sinon - mettre le switch flottant à 1 ;  
 - aller à FLOT 4 ;

FLOT 3.9 - test du switch point  
 si  $\neq$  0 - mettre le switch flottant à 1  
 - aller à FLOT 4 ;  
 sinon, mettre le switch point à 1 ;  
 aller à FLOT 3.4 ;

FLOT 3.10 - si le caractère = +, aller à FLOT 3.4 ;

FLOT 3.11 - si le caractère = -, aller à FLOT 3.4 ;

FLOT 3.12 - si le caractère = ., mettre le switch point à 1 ; aller à FLOT 3.4 ;

FLOT 3.13 - si le caractère = BLINK, mettre 1 dans switch flottant, aller à FLOT 4 ;

FLOT 3.14 - si le caractère est  $\neq$  de blanc, mettre le switch flottant à 1, aller à FLOT 4 ;

FLOT 4 - tester le switch flottant  
 si = 0, - transférer la zone dans OUTPUT ;  
 - ajouter la longueur de la zone dans CAR ;  
 - transformer la zone ASCII en BCD ;  
 - transférer la zone BCD dans CARD en la cadrant à gauche (générer des blancs éventuels à droite) ;  
 - aller à DEPAC 10 ;

si = 1, - transférer la zone dans OUTPUT avec un caractère BLINK en première position ;  
 - ajouter la longueur de la zone dans CAR ;  
 - mettre le switch erreur à 1 ;  
 - aller à DEPAC 10 ;

DEPAC 10 - tester si c'est la dernière zone d'INPUT  
 si oui, aller à DEPAC 11 ;  
 sinon - générer le caractère HT dans

OUTPUT ;  
- additionner 1 à CAR ;  
- passer à la variable suivante ;  
- aller à DEPAC 1 ;

DEPAC 11 - générer les caractères ETX et PR dans  
OUTPUT ;  
- additionner 2 à CAR ;  
- tester le switch d'erreur  
  si = 1, aller à DEPAC 12 ;  
  sinon fin de la routine ;

DEPAC 12 - charger CAR dans le deuxième mot de  
OUTPUT ;  
- exécuter la routine TALK ;  
- aller à DEPACK ;

FIN - fin.

11. Routine d'attente (WAIT)- But

Le but de cette routine est d'attendre un temps donné.

- Instruction d'appel et arguments

CALL WAIT (ITIME)

ITIME : temps d'attente, en secondes.

- Algorithme

Appel à la routine GEWAKE.



12. Routine de déconnexion d'un terminal (DISCON)- But

Cette routine a pour but de déconnecter un terminal.

- Instruction d'appel et arguments

CALL DISCON (ITERM)

ITERM - station code du terminal.

- Algorithme

PAS 1 - mettre le station code (ITERM) dans la VFD  
de la GEROUT 17 ;

PAS 2 - exécuter la GEROUT 17 ;

PAS 3 - fin de la routine.

### 5.1.5 Description des grilles

#### 5.1.5.1 Types de cartes

Nous avons utilisé un système de description de grilles par cartes ainsi qu'un programme de tri et chargement des cartes sur disque constituant le fichier des grilles.

##### Carte de type 1

=====

Contenu : numéro de grille  
numéro de carte  
type de carte = 1  
ligne et colonne  
déclaration de libellés et de variables

- les libellés sont donnés tels qu'ils apparaîtront à l'écran
- les variables sont décrites avec leur type, numérique ou alphanumérique, et leur longueur.

##### Carte de type 6

=====

Contenu : numéro de grille  
numéro de carte  
type de carte = 6

En général, une grille s'affiche à l'écran après avoir effacé le contenu précédent de cet écran. Ceci est dû au caractère de contrôle (FORM FEED) que la grille contient. La carte de type 6 a pour but de supprimer ce caractère et permet ainsi l'enchaînement de deux grilles sans effacer la première.

##### Carte de type 7

=====

Ce type de carte est associé à une déclaration de variable.

Contenu : numéro de grille  
numéro de carte  
type de carte = 7  
ligne et colonne  
spécification du format de la variable  
de la longueur  
de la présence obligatoire  
de fourchettes éventuelles

./...

### Format et longueur de la variable

AW variable alphanumérique de W caractères maximum  
 IW variable entière de W caractères maximum  
 FWD variable flottante de W caractères maximum dont  
 d décimales

### Fourchettes

Les fourchettes (éventuelles) sont exprimées sous la forme : valeur minimum : valeur maximum (les extrêmes sont inclus). Pour une même variable, les fourchettes sont séparées par des slashes (/).

### Carte de type 8

=====

Contenu : numéro de grille  
 numéro de carte  
 type de carte = 8  
 nom de l'application en 8 caractères.

Cette carte permet de spécifier le nom de l'application qui sera repris dans la table d'index du fichier des grilles et qui permettra par adjonction du USER-ID et du mot de passe de retrouver la grille sur le fichier ou d'afficher le menu d'un utilisateur déterminé.

Afin de donner plus de souplesse au gestionnaire de grilles, les types 6, 7 et 8 ont été ajoutés à ce qui existait.

### 5.1.5.2 Fichier des grilles

Un programme de chargement trie les cartes et constitue le fichier des grilles, en effectuant quelques contrôles de format.

Contenu du fichier (pour chaque numéro de grille) :

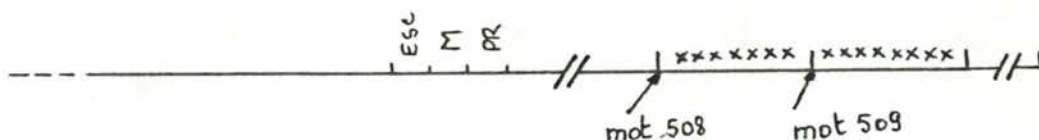
- grille	512 mots	8 secteurs	ASCII
- format	64 mots	1 secteur	BCD
- fourchettes	448 mots	7 secteurs	BCD

### GRILLE

=====

N° Grille	#c	TG	ESC	N	FFD	DEL	ESC	ESC2	ADREM	LIGNE	COLONNE	FS	TEXTE	G-S	Type de variable
-----------	----	----	-----	---	-----	-----	-----	------	-------	-------	---------	----	-------	-----	------------------





N° grille : numéro de grille indiqué sur la carte  
identifie une grille sur le fichier  
des grilles

#C : nombre de caractères de la grille à partir  
du troisième mot jusqu'à PR y compris

TG : type de grille : 0, 1, -1

Caractères de contrôles :

ESC : changement de mode

N : mode normal

FFD : - efface l'écran et positionne le  
curseur à la première position  
- efface les zones variables et posi-  
tionne le curseur sur la première  
variable

DEL : caractère de remplissage

ESC2 : suppression de toutes les tabulations

ADREM : indique que les deux caractères sui-  
vants sont le numéro de ligne et le  
numéro de colonne

FS : - début zone fixe

- fin zone variable

GS : début zone variable

M : mode format

PR : positionne le curseur à la première  
variable

LIGNE : numéro de ligne

COLONNE : numéro de colonne

TEXTE : libellés qui apparaîtront à l'écran

Type variable : type de variable donné par les cartes  
de type 1

cette zone est suivie par x blancs,  
x étant la longueur de la variable

Les mots 508 et 509 sont chargés par une carte de type  
8. Ils donnent le nom de l'application.

FORMAT

=====

Chaque variable possède un format.

Les formats apparaissent dans la zone FORMAT dans le  
même ordre que les variables apparaissent dans la zone  
GRILLE. Ils sont séparés par des virgules et délimités  
par des parenthèses.

Ex. : (I3, A5, F8.3, A1)

./...

#### FOURCHETTES =====

Les éventuelles fourchettes sont, comme les formats, décrites dans l'ordre des variables. Elles sont séparées par des virgules. Lorsqu'elles sont omises pour une variable, deux virgules se suivent.

Ex. : O:999,AAAAA/BBBBB,,←A/F/S,  
avec ← pour zone obligatoire

#### 5.1.5.3 Table d'index

Chaque élément de la table d'index du fichier grille a la structure suivante :

USER-ID : en 8 caractères  
mot de passe : en 8 caractères  
application : en 8 caractères  
n° grille : en 4 caractères  
adresse grille : en 4 caractères

Le USER-ID, le mot de passe et l'application sont introduits par des cartes. Le numéro de grille et l'adresse de la grille sont déterminés par le programme de constitution de la table d'index, en consultant le fichier des grilles.



## 5.2 Description des données

-----

Il faut remarquer que nous n'avons pas modifié le contenu d'un article. Les zones d'un même article gardent le même format et la même signification.

A chaque article, nous fait correspondre un dessin d'écran, c'est-à-dire une grille. Nous avons donc cinquante-quatre grilles. Le nombre peut paraître énorme d'autant plus que certains articles ont presque la même structure. Nous avons cependant remarqué qu'une standardisation des grilles et ainsi donc une réduction de leur nombre, aurait entraîné plusieurs inconvénients pour l'utilisateur : l'absence d'intitulé pour chaque article et un manque de clarté dans les libellés désignant les zones d'un article. Par exemple, prix d'achat, prix de revient, prix de vente ou prix de cession, beaucoup plus explicites que prix, apportent une aide notable à l'utilisateur, ainsi que l'intitulé complet de l'article sur lequel s'effectue la transaction.

Dans le système que nous avons réalisé, pour mettre à jour un enregistrement de la base de données, l'utilisateur doit introduire par écran, les éléments suivants :

- le code fonction
- le générique
- l'identifiant de l'enregistrement qui est soit la codification si c'est un article codé, soit le code émetteur, le code récepteur et le type de relation s'il s'agit d'un article relation.

Ces éléments étaient également requis dans le système existant. Vu qu'un enregistrement peut se trouver entièrement sur un écran, le code séquence est donc supprimé. Bien sûr, le code carte est devenu superflu.

Sur l'écran, qui possède vingt-quatre lignes de quatre-vingts positions, le code fonction, le générique et l'identifiant occupent trois lignes, le reste de l'écran étant disponible pour les données. Un article possédant au maximum trois lignes de données, nous les avons encadrées par un tableau à l'intérieur duquel chaque variable est accompagnée de son libellé.

A l'intérieur de la grille, un curseur parcourt toutes les zones variables. Lorsque la zone variable n'est pas complètement remplie, l'utilisateur doit frapper la touche "TAB" pour passer à la suivante. Nous avons disposé les zones à l'intérieur d'un article de manière à avoir les zones les plus utilisées en début de

./...



tableau pour éviter une série de commandes "TAB". En effet, l'utilisateur peut renvoyer l'article sans avoir nécessairement parcouru toutes les zones variables. Il faut également remarquer que certaines zones sont associées à d'autres. Par exemple, un prix est toujours accompagné d'un coefficient d'évolution et d'une date de saisie. Dans la grille que nous avons dessinée, ces trois zones ne sont donc pas séparées. Les cinquante-quatre grilles ont, dans la mesure du possible, le même schéma. Dans les différentes lignes de chaque article, on retrouve presque toujours les mêmes types de données. Les différents dessins d'écran sont présentés dans l'annexe.

## 5.3 Description des traitements

---

### 5.3.1 Description générale

Dans la procédure de mise à jour du système réalisé, l'utilisateur est en liaison permanente avec la banque de données. Chaque transaction effectue une mise à jour d'un enregistrement de la banque de données. Une transaction se termine lorsque la mise à jour est complètement et correctement réalisée. En effet, grâce au dialogue soutenu par le gestionnaire de grilles et par les programmes d'application, l'utilisateur prend connaissance de ses erreurs et est tenu de les corriger au fur et à mesure. Il s'agit d'erreurs de fourchettes et de formats des zones, détectées par le gestionnaire de grille, et d'erreurs détectées par les programmes d'application qui n'assurent plus que les contrôles croisés (contrôles de cohérence entre zones d'un article et entre articles).

L'utilisateur peut à tout moment mettre à jour n'importe quel article avec n'importe quel code fonction. Il peut donc travailler à son propre rythme sans bien entendu pouvoir modifier ou supprimer un enregistrement qu'il n'a pas encore créé. Cependant, il est préférable d'effectuer ensemble toutes les mises à jour concernant les articles d'un même type. Le code étant décomposé en plusieurs variables et rappelé sur l'écran lorsqu'on passe d'un article à un autre du même type (et pour une même fonction), l'utilisateur peut en effet ne changer qu'une partie du code. Ceci est surtout intéressant dans le cas des standards dont l'identifiant est composé de vingt-neuf positions (quatorze positions pour le code émetteur, quatorze pour le code récepteur et une pour le type de relation).

Les programmes de l'application réalisée sont fortement allégés par rapport aux programmes de l'application existante en ce sens que :

- tous les programmes qui mettent en forme les cartes de données (LECTUR et TRIDON) avant la mise à jour de la base de données deviennent inutiles.
- les contrôles syntaxiques et de fourchettes sont effectués par le gestionnaire de grilles

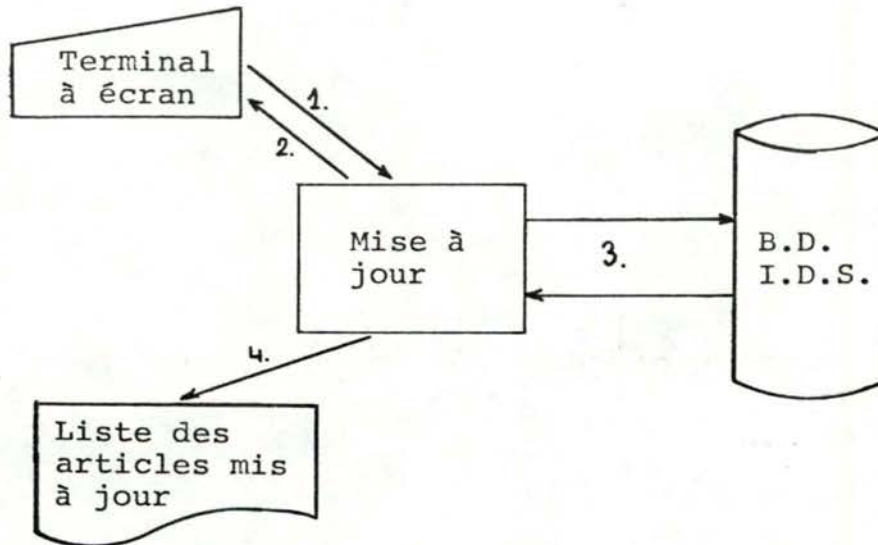
Dans cette application, en plus des fonctions de mise à jour implémentées dans le système existant

./...

(création, modification et suppression), nous avons prévu une fonction supplémentaire de consultation d'un enregistrement ainsi qu'une fonction d'appel de la liste des génériques.

Chaque transaction effectuée correctement est imprimée afin que l'utilisateur reçoive une liste de ses mises à jour.



5.3.2 Graphe\_général

1. Introduction des données
2. Affichage de l'article demandé et des éventuels messages d'erreur
3. Accès aux articles IDS
4. Suppression de la transaction effectuée

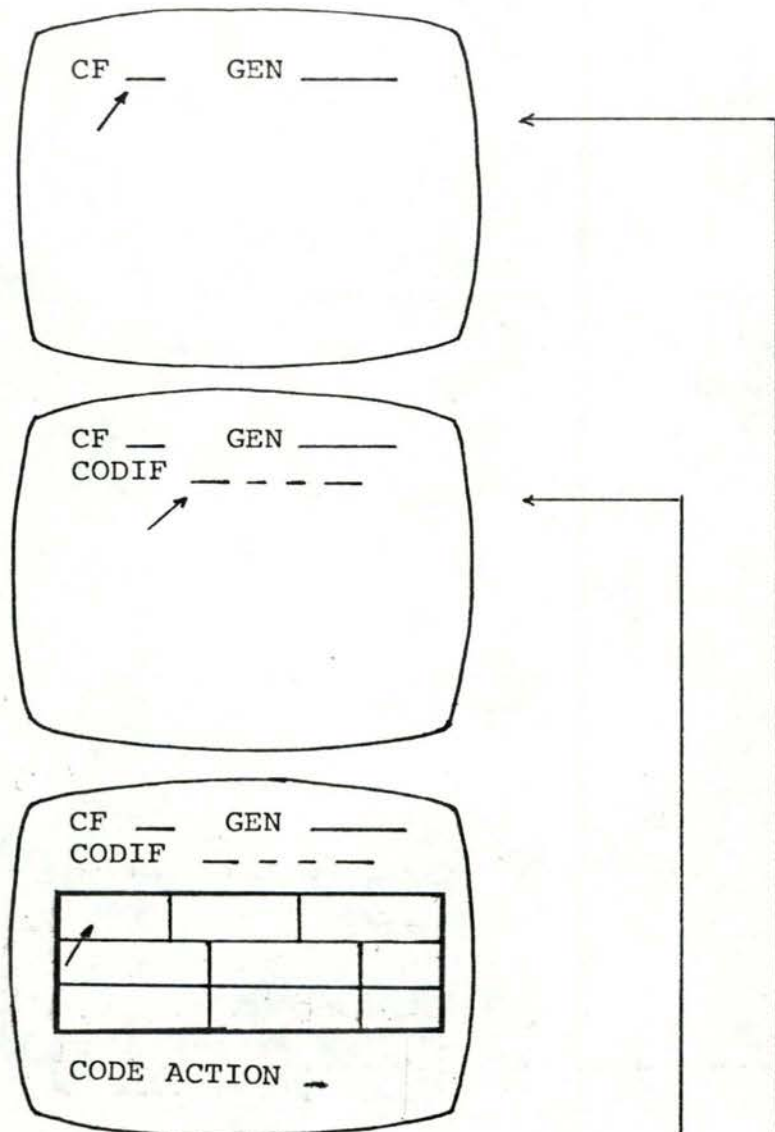
### 5.3.3 Description d'une transaction dans le système proposé

Dans le nouveau système, l'introduction des informations nécessaires au bon déroulement d'une transaction est effectuée en plusieurs étapes :

- en remplissant une première grille, l'utilisateur doit obligatoirement introduire le code fonction et le générique de l'article.
- ce générique, identifiant le type d'article, permet l'enchaînement automatique d'une deuxième grille associée à ce type d'article. Cette deuxième grille demande à l'utilisateur de fournir la codification de l'enregistrement. Elle rappelle aussi les données introduites dans la première grille
- la codification identifie l'occurrence de l'article à mettre à jour. Une troisième grille peut, une fois la codification introduite, être envoyée automatiquement à l'utilisateur qui y retrouvera tout le contenu de la précédente (soit tous les éléments introduits depuis le début de la transaction), les zones de données de l'enregistrement en provenance de la base de données ainsi qu'une zone appelée "code action". Cette dernière zone que l'utilisateur va compléter permet de décider de la transaction suivante.

Dans chacune de ces trois grilles, chaque variable à introduire est précédée d'un libellé explicite.

Graphe d'enchaînement des grilles



CODE ACTION : 1 } \_\_\_\_\_  
 N } \_\_\_\_\_  
 2 \_\_\_\_\_  
 F → Fin des mises à jour

./...



Après avoir montré la nécessité d'une démarche en trois étapes, nous allons étudier la description d'une transaction en détails.

Préalablement à la première mise à jour, le programme d'application réalise plusieurs fonctions :

- ouverture de la banque de données
- ouverture du fichier de sortie où les mises à jour sont imprimées
- accès à l'article de tête (HEADER) par la fonction IDS IRTRIM (fonction d'accès à un article codé sur base de son identifiant et transfert de cet enregistrement en mémoire)
  - pour tester si la base de données existe
  - pour tester si une simulation est en cours
  - pour mettre à jour la date de dernière mise à jour
- demande à l'utilisateur, par l'intermédiaire d'une grille, s'il désire ou non sauver la banque de données avant de commencer ses mises à jour

#### Au niveau de la première grille

Les différentes valeurs possibles que l'utilisateur peut introduire sont les suivantes :

Code fonction  
=====

C	création
M	modification
S	suppression
L	consultation du contenu d'un article (Lecture)
H	consultation de la liste des génériques (Help)

Ces deux dernières fonctions ont été ajoutées dans le but d'apporter une aide supplémentaire à l'utilisateur.

#### La fonction de lecture

L'utilisateur peut, à tout moment, avoir besoin de consulter un article. Par exemple, avant de créer une nouvelle occurrence de l'article "sous-domaine de vente", il peut vouloir consulter l'occurrence de l'article "Domaine de vente", hiérarchiquement supérieur, pour connaître le code de l'unité courante afin de garder la concordance entre ces deux enregistrements.

./...

### La fonction help

En cas d'oubli de la codification d'un générique, l'utilisateur a la possibilité de faire apparaître à l'écran la liste des codifications.

#### Générique =====

Les valeurs du générique sont les mêmes que celles qui ont été citées dans la présentation du système existant.

A ce niveau, les programmes d'application effectuent les contrôles de cohérence (contrôles croisés) sur le code fonction et le générique et envoient les messages suivants à l'utilisateur :

<u>CF</u>	<u>GEN</u>	<u>MESSAGE D'ERREUR</u>
S	HEAD	Il est interdit de supprimer le header
C	HEAD	Il est interdit de créer le header
S	DPTU	Il est interdit de supprimer un DPTU
M	DPTU	Il est interdit de modifier un DPTU
C	DPTU	Il est interdit de créer un DPTU
S	SDOU	Il est interdit de supprimer un SDOU
M	SDOU	Il est interdit de modifier un SDOU
C	SDOU	Il est interdit de créer un SDOU
C		Veillez donner le générique
S		Veillez donner le générique
M		Veillez donner le générique
L		Veillez donner le générique

Si le code fonction est H, le générique n'est pas contrôlé.

Si l'article de tête (HEAD) n'existe pas, c'est-à-dire si la banque de données n'a pas été créée, aucune mise à jour ne peut évidemment avoir lieu, si ce n'est la création du header.

La page suivante présente le dessin de la première grille, comme il apparaîtra à l'écran.

./...







### Au niveau de la deuxième grille

C'est à ce niveau que l'identifiant d'un article est introduit. Il s'agit soit de la codification si c'est un article codé ; soit du code émetteur, du code récepteur et du type de relation si c'est un article relation.

Nous remarquons que l'article de tête ne possède pas de codification, puisque ce type d'article ne compte qu'une seule occurrence. Cette deuxième grille est donc omise lors d'une mise à jour du HEADER.

L'identifiant, hiérarchique, est composé des codes des éléments supérieurs. Certaines positions ne servent qu'à aligner toutes les codifications sur une longueur maximale de quatorze positions. Ces nombreuses zones de complétage se situent en fin de codification, ou même sont insérées entre des éléments significatifs.

Dans le système existant, les quatorze positions devaient nécessairement être toutes introduites.

Dans le système proposé, seulement les éléments significatifs de la codification sont demandés. Chacun est isolé et est précédé d'un libellé. Cette démarche permet à l'utilisateur de mieux situer dans la banque de données l'article qu'il veut mettre à jour. De plus, il n'est pas obligé d'introduire les zones constantes qui apparaissent à l'écran, ni les zones de complétage, qui sont générées par le programme d'application avant d'exécuter les fonctions IDS.

Nous présentons dans le tableau suivant (5.1), pour chaque type d'article, les différentes parties de la codification et les contrôles effectués sur chacune d'entre elles (contrôles de présence et de numéricité)

./...

TABLEAU 5.1

	Longueur	Numérique	Obligatoire
USIN 2			
usine	1	x	x
DOVE 3			
domaine	2		x
SOCO 4			
n° société	1		x
TYFP 5			
type fourniture primaire	2		x
DEPT 6			
usine	1	x	x
département	1	x	x
SDVE 7			
domaine	2		x
sous-domaine	1		x
MARC 8			
société commercia- lisation	1		x
marché	3		x
SDCR 9			
domaine	2		x
sous-domaine	1		x
domaine critère	1		x
FPTO 10			
type fourniture primaire	2		x
nature	3		x
fournisseur	7		

./...

	Longueur	Numérique	Obligatoire
BLOC 11			
usine	1	x	x
département	1	x	x
bloc	2	x	x
M-CL 12			
société commercia- lisation	1		x
marché	3		x
client	3		x
ENTX 13			
fournisseur	7		x
DIRG 15			
usine = 1			
bloc	2		x
SKFP 16			
usine	1	x	x
département	1	x	
type fourniture primaire	2		x
nature	3		x
S-AD 17			
usine	1	x	x
département	1	x	
section	3		x
S-AU 18			
usine	1	x	x
département	1	x	
section	3		x

./...



	Longueur	Numérique	Obligatoire
S-PR 19			
usine	1	x	x
département	1	x	x
bloc	2	x	x
section	3		x
FLPR 20			
usine	1	x	x
département	1	x	x
bloc	2	x	x
section	3		
produit	7		x
SKPR 21			
usine	1	x	x
département	1	x	x
bloc	2	x	x
filrière	2		
produit	7		x
SKCH 22			
domaine	2		x
sous-domaine	1		x
domaine critère	1		x
autres critères	3		x
origine/destination	7		
S-VE 23			
usine = 0			
département = 9			
section	3		x

./...

	Longueur	Numérique	Obligatoire
PMCL 24			
domaine	2		x
sous-domaine	1		x
domaine critère	1		x
autres critères	3		
marché-client	7		x
S-DG 25			
usine = 1			
bloc	2		x
section	3		x
FPUS 26			
usine	1	x	x
département	1	x	
type fourniture primaire	2		x
nature	3		x
fournisseur	7		
CDEP 27			
usine	1	x	x
département 1	1	x	x
bloc	2	x	x
filière	2		
département 2	1	x	x
produit	7		x
ECHX 28			
domaine	2		x
sous-domaine	1		x
domaine critère	1		x
autres critères	3		x
fournisseur	7		x
Récepteur :			
générique : SKCH			
code : cfr 22			

	Longueur	Numérique	Obligatoire
SQA1 29 Emetteur : S-AU cfr 18 Récepteur : S-AD cfr 17 type de relation (= " 2" , " 3")	2		x
SQA2 30 Emetteur : S-AU cfr 18 Récepteur : S-AU cfr 18 type de relation (= " 1" , " 2" , " 3")	2		x
SQA3 31 Emetteur : S-AU cfr 18 Récepteur : S-PR cfr 19 type de relation = (" 1" , " 2" , " 3")	2		x
SQSP 32 Emetteur : S-PR cfr 19 Récepteur : FLPR cfr 20 type de relation = " 1"			



	Longueur	Numérique	Obligatoire
SQSV 33			
Emetteur : S-VE cfr 23			
Récepteur : PMCL cfr 24			
type de relation = " 1"			
FDLC 34			
Domaine	2		x
Sous-domaine	1		x
Domaine critère	1		x
Flux	10		x
Emetteur : SKCH cfr 22			
Récepteur : SKCH cfr 22			
PLIE 35			
Emetteur : FLPR cfr 20			
Récepteur : SKPR cfr 21			
DPP 36			
Emetteur : FLPR cfr 20			
Récepteur : SKPR cfr 21			
CVEN 37			
Emetteur : FLPR cfr 20			
Récepteur : SKCH cfr 22			

	Longueur	Numérique	Obligatoire
MIX 38 Emetteur : SKCH cfr 22 Récepteur : PMCL cfr 24			
SQN1 39 Emetteur : SKFP cfr 16 Récepteur : S-AD cfr 17 type de relation = (" 2" , " 3")	2		x
SQN2 40 Emetteur : SKFP cfr 16 Récepteur : S-AU cfr 18 type de relation = (" 1" , " 2" , " 3")	2		x
SQN3 41 Emetteur : SKFP cfr 16 Récepteur : S-PR cfr 19 Type de relation (= " 1" , " 2" , " 3")	2		x
SQN4 42 Emetteur : SKFP cfr 16 Récepteur : FLPR cfr 20 type de relation (= " 1")			

./...

	Longueur	Numérique	Obligatoire
SQP1 43			
Emetteur : SKPR cfr 21			
Récepteur : FLPR cfr 20			
type de relation (= " 1")			
SQP2 44			
Emetteur : CDEP cfr 27			
Récepteur : FLPR cfr 20			
type de relation (= " 1")			
CACH 45			
Emetteur : SKCH cfr 22			
Récepteur : SKPR cfr 21			
FPVE 46			
usine = 0			
département = 9			
type fourniture primaire	2		x
nature	3		x
fournisseur	7		
FPDG 47			
usine = 1			
type fourniture primaire	2		x
nature	3		x
fournisseur	7		

./...



	Longueur	Numérique	Obligatoire
SQN5 48			
Emetteur : FPVE cfr 46			
Récepteur : S-VE cfr 23			
type de relation (= " 1" , " 2" , " 3")	2		x
SQN6 49			
Emetteur : FPVE cfr 46			
Récepteur : PMCL cfr 24			
type de relation (= " 1" , " 4")	2		x
SQN7 50			
Emetteur : FPDG cfr 47			
Récepteur : S-DG cfr 25			
type de relation (= " 2" , " 3")	2		x
DPTU 51			
usine	1	x	x
département	1	x	x
SDOU 52			
domaine	2		x
sous-domaine	1		x
CSPE 53			
code	14		x

./...

	Longueur	Numérique	Obligatoire
ELCO 54			
Emetteur : CSPE cfr 53 (reprendre le code de la contrainte à la- quelle se rapporte cet élément)			
générique et code de l'élément de la contrainte			
- générique (valeurs pos- sibles : voir liste des gé- nériques)	4		x
- code	14		x

Une fois la codification validée, le programme d'appli-  
cation peut accéder à l'occurrence de l'article.

#### En cas de création

- s'il s'agit d'un article codé, le programme d'ap-  
plication recherche d'abord l'occurrence du maître  
de l'article, par la fonction IDS IRTRI (fonction  
d'accès à un article codé sur base de son identi-  
fiant).

Cette recherche a déjà été faite pour les articles  
des types suivants : USIN, DOVE, SOCO, TYFP, ENTX,  
DIRG et CSPE. En effet, ces articles ont pour  
maître l'article de tête, qui a déjà été recherché  
en début de traitement.

- s'il s'agit d'un article relation, le programme  
d'application recherche les occurrences des deux  
maîtres, par deux fonctions IDS IRTRIM (fonction  
d'accès à un article codé sur base de son identi-  
fiant et transfert de cet enregistrement en mémoire)

#### Dans les autres cas (modification, suppression, consulta- tion)

- s'il s'agit d'un article codé.  
Recherche de l'enregistrement par la fonction IDS  
IRTRIM.

./...

- s'il s'agit d'un article relation.

Recherche de l'enregistrement par la fonction IDS IRTISM (fonction d'accès à un article relation sur base de son identifiant et transfert de cet enregistrement en mémoire).

Après l'exécution d'une routine IDS, le programme d'application vérifie si elle s'est bien déroulée. En cas d'erreur, le module IDS transmet un message au programme d'application qui l'affiche sur la dernière ligne de l'écran.

Les deux dessins de grilles suivants présentent respectivement, à titre d'exemple, la grille de codification d'une section de production et la grille de codification d'un standard de consommation d'une section auxiliaire par une section administrative. Sur chacun de ces deux dessins, la première ligne rappelle le code fonction et le générique. Les zones, n'étant pas parcourues à ce niveau par le curseur, sont inaccessibleles à l'utilisateur.







<b>INFORMATIQUE</b> DEST. ORIG.	OBJET:	N° REF. <b>029</b> DATE: PAGE:
---------------------------------------	--------	--------------------------------------

CF X	GEN	SQA1	STANDARD CØNS	SECT	AUX	PAR	SECT	ADMINI	STR
S-AU :	USINE	N	DEPARTEMENT N	SECTION	XXX				
S-AD :	USINE	N	DEPARTEMENT N	SECTION	XXX				
TYPE RELATION XX									

### Au niveau de la troisième grille

A ce niveau, l'utilisateur peut mettre à jour les données de l'enregistrement qu'il a sélectionné. Le code fonction, le générique et la codification introduits aux niveaux précédents sont visualisés au début de la grille des données, comme constantes. En plus des données, l'utilisateur doit introduire un code action qui va déterminer la transaction suivante.

#### - Création

=====

- Affichage de la grille formattée avec les libellés
- Positionnement du curseur sur la première zone de variable
- Introduire des données
- Vérification des zones, y compris le code action (voir tableau 5.2)
  - 1) Contrôles de fourchettes exécutés par le gestionnaire de grilles. Les zones erronées clignotent.  
Correction de ces erreurs.
  - 2) Contrôles de cohérence effectués par le programme d'application. Des astérisques sont envoyés dans les zones erronées. Dans certains cas, envoi d'un message explicite.  
Correction de ces erreurs.
- Enregistrement de l'article dans la banque de données
- Analyse du code action :
  - si le code action vaut 1 : retour à la première grille. Le code fonction et le générique n'ont pas été mémorisé et l'utilisateur se trouve dans la même situation que lors de la première transaction.
  - si le code action vaut 2 : retour à la deuxième grille. Cette grille apparaît avec la codification de l'article traité dans la transaction précédente. L'utilisateur reste donc dans le même type de transaction (même code fonction) sur le même type d'article (même générique). Sans pouvoir changer la structure

./...



de la codification, il peut en changer la valeur en modifiant une ou plusieurs positions. Le fait de rappeler la codification lui permet de parcourir, en changeant très peu de positions, des occurrences d'un même article.

- si le code action vaut F : fin du traitement de mise à jour

#### - Modification

=====

- Affichage de la grille formatée avec les libellés et des données de l'article à modifier.
- Positionnement du curseur sur la première variable.
- L'utilisateur a la possibilité de modifier chaque zone ou de la garder.
- Vérification des zones (y compris le code action) :
  - 1) Contrôles de fourchettes exécutés par le gestionnaire de grilles. Les zones erronées clignotent.  
Correction de ces erreurs.
  - 2) Contrôles de cohérence effectués par le programme d'application. Des astérisques sont envoyés dans les zones erronées.  
Correction de ces erreurs.
- Modification de l'article dans la banque de données.
- Analyse du code action : comme dans le cas d'une création, le code action peut valoir 1, 2 ou F, avec la même signification.

#### - Suppression

=====

- Affichage de la grille formatée avec les libellés et des données de l'article à supprimer.
- Positionnement du curseur sur le code action uniquement.
- Analyse du code action :

./...

- si le code action vaut 1 : suppression de l'article dans la base de données et retour à la première grille.
- si le code action vaut 2 : suppression de l'article dans la base de données et retour à la deuxième grille.
- si le code action vaut F : suppression de l'article dans la base de données et fin du traitement de mise à jour.
- si le code action vaut N : l'enregistrement n'est pas supprimé ; retour à la première grille.

Nous avons voulu laisser à l'utilisateur la possibilité d'infirmer son ordre de suppression après avoir vérifié le contenu de l'article visualisé.

#### - Consultation

=====

- Affichage de la grille formatée avec les libellés et des données de l'article à consulter.
- Positionnement du curseur sur le code action uniquement.
- Analyse du code action : comme dans le cas d'une création et d'une modification, il peut prendre les valeurs 1, 2 ou F avec la même signification.

#### Contrôles des zones de données et messages d'erreurs

On trouve dans le tableau 5.2, la liste des différentes données reprises dans les articles. Pour chaque variable, nous donnons son format, les contrôles réalisés par le gestionnaire de grilles et ceux réalisés par le programme d'application.

Formats : AW zone alphanumérique de W caractères maximum  
 IW zone entière de W caractères maximum  
 FW.d zone flottante de W caractères maximum dont d décimales

Remarque : "positif" signifie strictement supérieur à zéro.

./...

TABLEAU 5.2

	Format	Gestionnaire de grilles	Programme d'application
<u>Libellé</u>	A30	obligatoire	<p>S'il s'agit d'un SDVE, le code unité doit être égal au code unité du DOVE correspondant.</p> <p><u>Envoi du message</u> : le code unité doit être égal au code unité du DOVE correspondant.</p> <p>S'il s'agit d'un SDCR, le code unité doit être égal au code unité du SDVE correspondant.</p> <p><u>Envoi du message</u> : le code unité doit être égal au code unité du SDVE correspondant.</p>
<u>Commentaire</u>	voir libellé		
<u>Unité courante</u>	A2	obligatoire	
- code			
- libellé	A6	obligatoire	
<u>Unité globale</u>	A2		
- code	A6		
- libellé			
<u>Unité</u>			
- code	A2	obligatoire	
- libellé	A6	obligatoire	



Unité d'oeuvre	Format	Gestionnaire de grille	Programme d'application
- code	A2	obligatoire	
- libellé	A6	obligatoire	
<u>Disponibilité</u>			
- mini	F16.5	numérique positif	} maxi > mini
- maxi	F16.5	numérique positif ou égal à - 1	
- date saisie	I6	numérique 010100 < date saisie < 311299	JJ ≤ 31 MM ≤ 12

Lorsque la date est omise alors qu'il y a des disponibilités, le programme d'application génère une valeur par défaut qui est la date du jour. Il faut remarquer que cette valeur par défaut n'étant pas considérée comme une erreur, ne provoque pas une nouvelle visualisation des données.

./...

/....

<u>Capacité en</u> <u>unité d'oeuvre</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Niveau de</u> <u>production</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Stock initial</u> <u>souhaité</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Stock final</u> <u>souhaité</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	

./...

<u>Quantité à céder/</u> <u>appro en mono-</u> <u>usine</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Quantité à</u> <u>déclasser</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Quantité à vendre</u> <u>modèle complet</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	
<u>Quantité à vendre</u> <u>modèle incomplet</u>	
- mini	} voir contrôles disponibilités
- maxi	
- date saisie	



Format	Gestionnaire de grilles	Programme d'aplication
<u>Prix ordre calculé</u> - prix ordre calculé - coefficient d'évolution - date de sai- sie du prix d'ordre cal- culé	F16.5 F8.3 I6 numérique positif numérique numérique 010100 \$ date saisie \$ 311299	interdit si pas de prix d'ordre calculé JJ \$ 31 MM \$ 12 interdite si pas de prix d'ordre calculé. Si la date de saisie est omise alors que le coût d'acquisition est présent, le programme d'application génère une valeur par défaut qui est la date du jour. Cette valeur par défaut ne provoque pas une nouvelle visualisa- tion des données.

/...

Format	Gestionnaire de grilles	Programme d'application
<u>Prix de cession</u> - prix de cession  - coefficient d'évolution  - date saisie du prix de cession	obligatoire positif numérique numérique  numérique  010100 \$ date saisie \$ 311299  voir Prix de cession	JJ \$ 31 MM \$ 12  Si la date de saisie est omise, le programme d'application génère une valeur par défaut qui est la date du jour
<u>Hypothèse de prix de revient du produit vendu</u>		

./...

	Format	Gestionnaire de grilles	Programme d'application
<u>Prix d'ordre</u> - prix d'ordre  - coefficient d'évolution  - date saisie du prix d'ordre	F16.5	numérique positif	Obligatoire si variation de stock c'est-à-dire si stock initial en quantité est différent du stock final souhaité (mini ou maxi)
	F8.3	numérique	interdit si pas de prix d'ordre
	I6	numérique 010100 < date saisie < 311299	JJ < 31 MM < 12  Lorsque la date est omise alors que le prix d'ordre est présent, le programme d'application génère une valeur par défaut qui est la date du jour. Cette valeur par défaut ne provoque pas une nouvelle visualisation des données
<u>Prix d'ordre en valeur</u>		voir prix d'ordre	

./...



	Format	Gestionnaire de grilles	Programme d'application
<u>Standard de consommation</u> - standard	F16.5	obligatoire numérique positif numérique	Le standard doit être compris entre les bornes de vraisemblance
- coefficient d'évolution - date saisie du standard	F8.3  I6	numérique  010100 \$ date saisie \$ 311299	JJ ≤ 31 MM ≤ 12
<u>Prix d'achat</u> <u>Prix de vente</u>		voir standard de consommation voir standard de consommation	Si la date de saisie est omise, le programme d'application génère une valeur par défaut qui est la date du jour.

Format	Gestionnaire de grilles	Programme d'application
<u>Pourcentage de dégradation</u> - pourcentage de dégradation - coefficient d'évolution - date de saisie du pourcentage	numérique  numérique  numérique  010100 ≤ date saisie ≤ 311299	interdit si pas de pourcentage de dégradation JJ ≤ 31 MM ≤ 12 interdite si pas de pourcentage de dégradation. Si la date de saisie est omise alors que le pourcentage de dégradation est présent, le programme d'application génère une valeur par défaut qui est la date du jour. Cette valeur par défaut ne provoque pas une nouvelle visualisation des données.

./...



Format	Gestionnaire de grilles	Programme d'application
<u>Coût d'acquisition</u> - coût d'acquisition - coefficient d'évolution - date saisie du coût d'acquisition	Positif Numérique numérique numérique 010100 < date saisie < 311299	Le coût d'acquisition doit être compris entre ses bornes de vraisemblance Interdit si pas de coût d'acquisition  JJ < 31 MM < 12  Lorsque la date est omise alors que le coût d'acquisition est présent, le programme d'application génère une valeur par défaut qui est la date du jour. Cette valeur par défaut ne provoque pas une nouvelle visualisation des données. La date de saisie est interdite si le coût d'acquisition est absent.



	Format	Gestionnaire de grilles	Programme d'application
<u>Date d'exploitation</u>	I6	numérique 010100 < date < 311299	JJ < 31 MM < 12
<u>Horizon de gestion</u>			
- date début	I6	numérique 010100 < date < 311299	JJ < 31 MM < 12
- date fin	I6	numérique 010100 < date < 311299	JJ < 31 MM < 12
<u>Page-Range</u>			
- mini	I6	numérique positif	
- maxi	I6	numérique positif	
<u>Indicateur d'édition</u>	I6	numérique valeurs possibles : 1 ou 0	

./...

	Format	Gestionnaire de grilles	Programme d'application
<u>Proportion</u> - proportion	F8.3	numérique positif obligatoire numérique 010100\$date saisie \$ 311299	doit être compris entre ses bornes  JJ \$ 31 MM \$ 12  Si la date de saisie est omise, alors le programme d'application génère une valeur par défaut qui est la date du jour.
- date saisie	I6		

./...

Coût de l'unité d'oeuvre sous- traitée	Format	Gestionnaire de grilles	Programme d'application
- coût de l'unité d'oeuvre sous- traitée	F16.5	numérique positif	obligatoire s'il s'agit d'une section auxiliaire sous-traitée c'est-à-dire si type = 3 ou 4. Dans ce cas, cette zone doit être comprise entre ses bornes de vraiseemblance. interdit s'il s'agit d'une section auxiliaire interne c'est-à-dire si le type = 1 ou 2.

./...



	Format	Gestionnaire de grilles	Programme d'application
<u>Contrainte</u> <u>marché</u> - mini - maxi - date saisie	F16.5	numérique positif obligatoire	maxi > mini JJ < 31 MM < 12 génération d'une valeur qui est la date du jour quand cette date de saisie est omise. Cette valeur par défaut ne provoque pas une nouvelle visualisation des données.
	F16.5	numérique positif obligatoire	
	I6	numérique 010100 < date saisie < 311299	

./....

	Format	Gestionnaire de grilles	Programme d'application
<u>Indicateur de cumul (pour FLPR)</u> - en unité courante	I1	numérique valeurs possibles : 0 ou 1 obligatoire	si l'indicateur de cumul en unité courante est égal à 1, alors l'unité courante du FLPR doit être égale à l'unité courante du BLOC (qui est le maître). Envoi du message : "Unité courante FLPR ≠ unité courante BLOC avec indicateur cumul unité courante = 1"
- en unité globale	I1	numérique valeurs possibles : 0 ou 1 obligatoire	Si l'indicateur de cumul en unité globale est égal à 1, alors l'unité globale du BLOC doit être différente de blanc. Envoi du message : "Unité globale BLOC = blanc alors que l'indicateur cumul en unité globale = 1"
<u>Coefficient de changement d'unité (FLPR)</u>	F13.6	numérique	doit être différent de 1 si l'indicateur de cumul en unité globale du FLPR = 1 et l'unité du FLPR est différente de l'unité globale du BLOC. Envoi du message : "Coefficient de changement d'unité = 1 avec indicateur de cumul en unité globale = 1 et unité FLPR ≠ unité globale BLOC."

./...

	Format	Gestionnaire de grilles	Programme d'application
<u>Indicateur de cumul (pour SKPR)</u> - en unité courante	I1	numérique valeurs possibles : 0 ou 1 obligatoire	si l'indicateur de cumul en unité courante est égal à 1, alors l'unité courante du SKPR doit être égale à l'unité courante du BLOC (qui est le maître) <u>Envoi du message</u> : "Unité courante SKPR ≠ unité courante BLOC avec indicateur de cumul en unité courante = 1"
- en unité globale	I1	numérique valeurs possibles : 0 ou 1 obligatoire	si l'indicateur de cumul en unité globale est égal à 1, alors l'unité globale du BLOC doit être différente de blanc. <u>Envoi du message</u> : "Unité globale BLOC = blanc alors que l'indicateur cumul en unité globale = 1"
<u>Coefficient de changement d'unité (SKPR)</u>	F13.6	numérique	doit être différent de 1 si l'indicateur de cumul en unité globale du SKPR = 1 et l'unité de SKPR est différente de l'unité globale du BLOC. <u>Envoi du message</u> : "Coefficient de changement d'unité = 1 avec indicateur de cumul en unité globale = 1 et unité SKPR ≠ unité globale BLOC."

./...



Format	Gestionnaire de grilles	Programme d'application
<u>Indicateur de cumul (PMCL)</u> - en unité courante	numérique obligatoire valeurs possibles : 0 ou 1	si l'indicateur de cumul en unité courante est égal à 1, alors l'unité courante du PMCL doit être égale à l'unité courante du SDCR (qui est le maître) Envoi du message : unité courante PMCL ≠ unité courante SDCR avec indicateur de cumul en unité courante = 1  si l'indicateur de cumul est égal à 1, alors l'unité globale du SDCR doit être différente de blanc. Envoi du message : unité globale SDCR = blanc alors que l'indicateur de cumul en unité globale = 1  doit être différent de 1 si l'indicateur de cumul en unité globale du PMCL = 1 et l'unité du PMCL est différente de l'unité globale du BLOC. Envoi du message : coefficient de changement d'unité = 1 avec indicateur de cumul en unité globale = 1 et unité PMCL ≠ unité globale SDCR.
- en unité globale	numérique obligatoire valeurs possibles : 0 ou 1	
<u>Coefficient de changement d'unité (PMCL)</u>	numérique	

	Format	Gestionnaire de grilles	Programme d'application
<u>Indicateur de cumul (pour CDEP)</u> - en unité courante	II	numérique valeurs possibles : 0 ou 1 obligatoire	si l'indicateur de cumul en unité courante est égal à 1, alors l'unité courante de la CDEP doit être égale à l'unité courante du BLOC (qui est le maître) <u>Envoi du message</u> : "unité courante CDEP ≠ unité courante BLOC avec indicateur de cumul en unité courante = 1"
- en unité globale	II	numérique valeurs possibles : 0 ou 1 obligatoire	si l'indicateur de cumul en unité globale est égal à 1, alors l'unité globale du BLOC doit être différente de blanc. <u>Envoi du message</u> : "unité globale BLOC = blanc alors que l'indicateur de cumul en unité globale = 1"
<u>Coefficient de changement d'unité (CDEP)</u>	F13.6	numérique	doit être différent de 1 si l'indicateur de cumul en unité globale du CDEP = 1 et l'unité du CDEP est différente de l'unité globale du BLOC. <u>Envoi du message</u> : "coefficient de changement d'unité = 1 avec indicateur de cumul en unité globale = 1 et unité CDEP ≠ unité globale BLOC"

./...



	Format	Gestionnaire de grilles	Programme d'application
<p><u>Indicateur de cumul (pour PLIE)</u></p> <p>- en unité courante</p>	<p>Il</p>	<p>numérique</p> <p>valeurs possibles : 0 ou 1</p> <p>obligatoire</p>	<p>si l'indicateur de cumul en unité courante du PLIE = 1, alors l'indicateur de cumul en unité courante du FLPR doit être = 0</p> <p><u>Envoi du message</u> : "Indicateur de cumul en unité courante PLIE et indicateur de cumul en unité courante = 1"</p> <p>si l'indicateur de cumul en unité courante est égal à 1, alors l'unité courante du PLIE doit être égale à l'unité courante du BLOC (qui est le maître)</p> <p><u>Envoi du message</u> : "unité courante PLIE ≠ unité courante BLOC avec indicateur de cumul en unité courante = 1"</p> <p>si l'indicateur de cumul en unité globale du PLIE = 1, alors l'indicateur de cumul en unité globale du FLPR doit être = 0</p> <p><u>Envoi du message</u> : "Indicateur de cumul en unité courante PLIE et indicateur de cumul en unité courante FLPR = 1"</p>
<p>- en unité globale</p>	<p>Il</p>	<p>numérique</p> <p>valeurs possibles : 0 ou 1</p> <p>obligatoire</p>	



Format	Gestionnaire de grilles	Programme d'application
<u>Coefficient de changement d'unité (PLIE)</u> - pour le BLOC	numérique	<p>si l'indicateur de cumul en unité globale est = à 1, alors l'unité globale du BLOC doit être différente de blanc.</p> <p><u>Envoi du message</u> : "unité globale BLOC = blanc alors que l'indicateur de cumul en unité globale = 1"</p> <p>doit être différent de 1 si l'indicateur de cumul en unité globale du PLIE = 1 et l'unité du FLPR est ≠ de l'unité globale du BLOC.</p> <p><u>Envoi du message</u> : "coefficient de changement d'unité = 1 avec indicateur de cumul en unité globale = 1 et unité FLPR ≠ unité globale BLOC"</p>

	Format	Gestionnaire de grilles	Programme d'application
- pour le stock	F13.6	numérique	doit être égal à 0 si l'unité du FLPR = unité SKPR doit être égal à 1 si l'unité du FLPR = unité SKPR
<u>Coefficient de changement d'unité (pour MIX)</u>	F13.6	numérique	si le coefficient de changement d'unité vaut 1, alors l'unité du SKCH doit être égale à l'unité du PMCL <u>Envoi du message : "coefficient de changement d'unité = 1, alors que unité SKCH ≠ unité PMCL"</u>

	Format	Gestionnaire de grilles	Programme d'application
<u>Indicateur de</u> <u>calcul du prix</u> <u>de revient pour</u> <u>PLIE</u>	A1	obligatoire valeurs possibles : 0 , 1 ou 2	si le produit boucle, c'est-à-dire si le numéro du bloc du SKPR est < au numéro du bloc du FLPR, l'indicateur de calcul du prix de revient doit = 1 Envoi du message : "indicateur de calcul ≠ 1 alors que numéro bloc du SKPR < numéro bloc FLPR"



Format	Gestionnaire de grilles	Programme d'application
<u>Section auxiliaire équivalente</u> - code	A14	obligatoire si la section auxiliaire possède une section auxiliaire équivalente c'est-à-dire si le type est égal à 2 ou 4 interdite si la section auxiliaire est sans équivalent c'est-à-dire si type est égal à 1 ou 3
<u>Durée d'activité en jours</u>	I5	numérique - 1 ≤ durée ≤ 99999
<u>Durée de simulation en jours</u>	I5	voir durée d'activité en jours

/...

	Format	Gestionnaire de grilles	Programme d'application
<u>Type de section auxiliaire</u>	I1	numérique obligatoire valeurs possibles : 1, 2, 3 ou 4	
<u>Type de flux</u>	I1	numérique obligatoire valeurs possibles : 1, 2, 3 ou 4	
<u>Type de con- trainte</u>	A2	obligatoire valeurs possibles : EQ , GE ou LE	

Format	Gestionnaire de grilles	Programme d'application
<u>Bornes de vrais-semblance</u> - mini	numérique	borne mini < borne maxi si la borne mini est omise, alors elle prend 0 comme valeur par défaut et la borne maxi prend comme valeur - 1 s'il s'agit du MIX, la borne mini doit être > 0 s'il s'agit du MIX, la borne maxi doit être < 100
- maxi	numérique	
<u>Coefficient de rendement</u>	numérique	
<u>Permission de delete</u>	obligatoire valeurs possibles : 0 ou 1	
<u>Indicateur d'édition</u>	obligatoire valeurs possibles : 0 ou 1	
<u>Indicateur de calcul du prix de revient pour DPP</u>	obligatoire valeurs possibles : 0 ou 1	si le produit boucle c'est-à-dire si le numéro du bloc du SKPR est < au numéro du bloc du FLPR, l'indicateur de calcul du prix de revient doit = 1 Envoi du message : "Indicateur calcul ≠ 1 alors que numéro bloc du SKPR < numéro bloc FLPR"



	Format	Gestionnaire de grilles	Programme d'application
<u>Prix tarif</u>			
- prix tarif	F16.5	numérique positif	doit être compris entre ses limites
- coefficient d'évolution	F8.3	numérique	interdit si pas de prix tarif
- date saisie du prix tarif	I6	numérique 010100 < date saisie < 311299	JJ < 31 MM < 12
			interdite si pas de prix tarif si la date de saisie est omise, alors que le prix tarif est présent, le programme d'application génère la date du jour comme valeur par défaut

./...

	Format	Gestionnaire de grilles	Programme d'application
<u>Part</u>	F16.5	numérique obligatoire positif 0 < part ≤ 100	doit être comprise entre ses bornes
<u>Terme de droite</u>	F16.5	numérique obligatoire	
<u>Coefficient de la matrice</u>	F16.5	numérique obligatoire	

Les deux dessins de grilles suivants représentent respectivement, pour poursuivre l'exemple, la grille des données d'une section de production et la grille des données d'un standard de consommation d'une section auxiliaire par une section administrative.

./...



INFORMATIQUE		OBJET:	N° REF.	019		
DEST.					DATE:	
ORIG.					PAGE:	

CF X GEN S-PR		SECTION DE PRODUCTION	
USINE N	DEPARTEMENT N	BLØC NN	SECTION XXX
LIBELLE	UNITE D'OEUVRE		
CODE	LIBELLE		
XXXXXX	XXXXXX	XXXXXX	XXXXXX
CAPACITE EN U.Ø.	DATE		
MINI	SAISIE		
MAXI	NNNNNN		
SNNNNNNN	NNNNNNNN	NNNN	NNNN
VRAI SEMBLANCE	PRIX REVIENT		
MINI	MAXI		
SNNNNNNN	SNNNNNNN	NNNN	NNNN
CODE ACTION X			



**OBJECT:**

1

—

5

REF.

029

DEST.

DATE :

ORIG.

PAGE :

CF X	GEN	SQA1	STANDARD CØNS :		SECT	AUX	PAR	SECT	ADMINISTR
S-AU :	USINE	N	DEPARTEMENT	N	SECTION	XXX			
S-AD :	USINE	N	DEPARTEMENT	N	SECTION	XXX			
TYPE RELATION XX									
STANDARD CØNSØMMATION SNNNNNNNN·NNNNN			CØEF EVØL SNN·NNN	DATE SAISIE NNNNNN	VRAISEMBLANCE MINI SNNNNNNNN·NNNNN		STANDARD MAXI SNNNNNNNN·NNNNN		DEL X
CØDE ACTION X									

#### 5.3.4 Algorithmes des programmes d'application

L'application de mise à jour est soutenue par plusieurs programmes :

- le programme principal (PP)
- 54 sous-programmes correspond à chacun des 54 types d'articles de la banque de données (PO1 à P54)
- un sous-programme d'impression des mises à jour (IMPERR)
- un sous-programme de visualisation de la liste des génériques (HELP)
- un sous-programme d'alignement de zone flottante sur le point décimal (FLOT).

##### Programme principal (PP)

=====

Le but de ce programme est de :

- traiter la grille du code fonction et du générique
- appeler le sous-programme qui correspond au générique
- analyser le code action

Le fichier accédé dans ce programme est la base de données

##### Algorithme

- PAS 1 Initialisation
- PAS 1.1 Ouverture de la banque de données ;
  - PAS 1.2 Ouverture du fichier d'impression (appel à IMPERR) ;
  - PAS 1.3 Recherche du header
    - PAS 1.3.1 Mise à jour de la date de dernière mise à jour ;
    - PAS 1.3.2 Si une simulation en cours ; aller à PAS 9 ;
- PAS 2 Recherche de l'identification du terminal par l'appel à la routine INQTER ;
- PAS 3 Chargement de la grille du code fonction et du générique en mémoire par l'appel à la routine GRILEC ;
- PAS 4 Introduction des données à l'écran par l'appel à la routine TALK ;

./...



- PAS 5 Contrôles croisés sur le code fonction et le générique.  
Si erreur, envoi d'un message d'erreur et aller à PAS 4  
sinon aller à PAS 6 ;
- PAS 6 Affichage du code fonction et du générique en tant que zones fixes par l'appel aux routines BLDFIX et SEND ;
- PAS 7 Appel du sous-programme correspondant au générique ;
- PAS 8 Test du code action  
si le code action est égal à 1, aller à PAS 4 ;  
si le code action est égal à n, aller à PAS 4 ;  
- si le code action est égal à F, aller à PAS 9 ;
- PAS 9 Terminaison  
PAS 9.1 Clôture de la banque de données  
PAS 9.2 Clôture du fichier d'impression

Sous-programme de traitement d'un article (Pxx)  
=====

Le but de ce sous-programme est de réaliser la création, la suppression, la modification et la consultation pour un type d'article, en traitant la codification et les données. Lors de l'introduction de la codification, l'utilisateur peut se tromper une fois avant que la grille du code fonction et du générique ne lui soit renvoyée.

Le fichier accédé dans ce sous-programme est la banque de données.

Algorithme

- PAS 1 Chargement en mémoire de la grille de codification (routine GRILEC) ;
- PAS 2 Initialisation du sous-statut à zéro (compteur du nombre d'essais pour l'introduction de la codification ;
- PAS 3 Introduction de la codification à l'écran (routine TALK) ;
- PAS 4 Test du code fonction (qui est transmis par le programme principal)  
Si le code fonction est égal à M, aller à PAS 5 ;  
Si le code fonction est égal à C, aller à PAS 6 ;  
Si le code fonction est égal à L, aller à PAS 7 ;  
Si le code fonction est égal à S, aller à PAS 8 ;

./...

## PAS 5 Traitement de modification

- PAS 5.1 Recherche de l'article
- PAS 5.2 Tester si l'article existe
  - si oui, aller à PAS 5.4 ;
  - sinon aller à PAS 5.3 ;
- PAS 5.3 Tester le sous-statut
  - si 2 - mettre 1 dans le code action ;
  - aller à PAS 10 ;
  - sinon - envoyer le message IDS à l'écran (routine INSTXT et SEND) ;
  - ajouter 1 au sous-statut ;
  - envoyer la codification erronée à l'écran ;
  - aller à PAS 3 ;
- PAS 5.4 - Affichage de la codification en tant que zone fixe (routine BLDFIX et SEND) ;
  - Chargement en mémoire de la grille des données (routine GRILEC) ;
- PAS 5.5 - Affichage à l'écran de la grille et du contenu de l'article ;
  - Introduction des modifications et du code action ;
  - contrôles croisés des données ;
  - si erreur, aller à PAS 5.6 ;
  - sinon, aller à PAS 5.7 ;
- PAS 5.6 - mettre des \* dans les zones erronées ;
  - aller à PAS 5.5 ;
- PAS 5.7 - modifier l'article dans la banque de données ;
  - impression de la mise à jour (appel à IMPERR) ;
  - aller à PAS 9 ;

## PAS 6 Traitement de création

- PAS 6.1 - recherche du ou des maîtres de l'article ;
- PAS 6.2 - tester si le(s) maître(s) existe(nt)
  - si oui, aller à PAS 6.4 ;
  - sinon, aller à PAS 6.3 ;
- PAS 6.3 - tester le sous-statut
  - si > 2 - mettre 1 dans le code action ;
  - aller à PAS 10 ;
  - sinon - envoyer le message IDS à l'écran ;
  - ajouter 1 au sous-statut ;
  - envoyer la codification erronée à l'écran ;
  - aller à PAS 3 ;

./...



- PAS 6.4 - Affichage de la codification en tant que zone fixe ;  
- chargement en mémoire de la grille des données ;
- PAS 6.5 - envoi à l'écran de la grille des données ;  
- introduction des données et du code action ;  
- contrôles croisés des données ;  
  si erreur, aller à PAS 6.6 ;  
  sinon, aller à PAS 6.7 ;
- PAS 6.6 - mettre des dans les zones erronées ;  
- aller à PAS 6.5 ;
- PAS 6.7 - stockage de l'article dans la banque de données ;  
- impression de la mise à jour ;  
- aller à PAS 9 ;

#### PAS 7 Traitement de consultation

- PAS 7.1 - Recherche de l'article
- PAS 7.2 - Tester si l'article existe  
  si oui, aller à PAS 7.4 ;  
  sinon, aller à PAS 7.3 ;
- PAS 7.3 - Tester le sous-statut  
  si 2 - mettre 1 dans le code action ;  
          - aller à PAS 10 ;  
  sinon - envoyer le message IDS à l'écran ;  
          - ajouter 1 au sous-statut ;  
          - envoyer la codification erronée à l'écran ;  
          - aller à PAS 3 ;
- PAS 7.4 - Affichage de la codification en tant que zone fixe ;  
- chargement en mémoire de la grille des données ;  
- affichage à l'écran de la grille et du contenu de l'article ;  
- introduction du code action ;  
- aller à PAS 9 ;

#### PAS 8 Traitement de suppression

- PAS 8.1 - Recherche de l'article ;
- PAS 8.2 - Tester si l'article existe  
  si oui, aller à PAS 8.4 ;  
  sinon, aller à PAS 8.3 ;
- PAS 8.3 - Tester le sous-statut  
  si 2 - mettre 1 dans le code action ;  
          - aller à PAS 10 ;  
  sinon - envoyer le message IDS à l'écran ;

./...



- ajouter 1 au sous-statut ;
- envoyer la codification erronée à l'écran ;
- aller à PAS 3 ;

- PAS 8.4 - Affichage de la codification en tant que zone fixe ;
- chargement en mémoire de la grille des données ;
  - affichage à l'écran de la grille et du contenu de l'article ;
  - introduction du code action ;
  - si l'article ne peut être supprimé, envoyer un message à l'écran ; aller à PAS 10 ;
  - suppression de l'article dans la banque de données ;
  - impression de la mise à jour ;
  - aller à PAS 9 ;

PAS 9 - tester le code action

- PAS 9.1 - si = 2, aller à PAS 9.2 ;  
sinon aller à P10 ;

- PAS 9.2 - remettre le sous-statut à zéro ;
- envoyer à l'écran la codification ;
  - aller à PAS 3 ;

PAS 10 - fin du sous-programme ;

Sous-programme d'impression des mises à jour (IMPERR)  
=====

Le but de ce sous-programme est d'imprimer les mises à jour qui se déroulent correctement dans l'application, et ainsi de fournir à l'utilisateur une trace de ses mises à jour. Il imprime une à trois lignes par article mis à jour (code fonction, codification et données)

Sous-programme de visualisation de la liste des génériques (HELP)  
=====

Le but de ce sous-programme est d'afficher à l'écran, à la demande de l'utilisateur, la liste des génériques.

#### Algorithme

PAS 1 - Chargement en mémoire de la grille des génériques ;

PAS 2 - Affichage de la grille à l'écran ;

./...

PAS 3 - Attendre 10 secondes ;

PAS 4 - Fin du sous-programme ;

Sous-programme d'alignement (FLOT)

=====

Le but de ce sous-programme est d'aligner une zone flottante sur le point décimal. Une variable flottante introduite à l'écran se trouve, après la routine de dépackage, dans la zone CARD où elle est cadrée à gauche et complétée éventuellement par des blancs. Cette zone est alphanumérique. Elle ne peut être traitée par des instructions arithmétiques. Le sous-programme transfère la zone dans une variable numérique (variable C) en la cadrant sur le point décimal. Aucun contrôle n'est à effectuer.

#### Algorithme

PAS 1 - Initialisation

PAS 1.1 - mettre 1 dans le switch de signe (ce switch vaudra 1 si la variable est positive et - 1 si elle est négative) ;

PAS 1.2 - mettre la variable A à zéro (A recevra la valeur qu'aurait la zone si le point était à droite du dernier chiffre) ;

PAS 1.2 - mettre la variable B à zéro (B sera égal à 10 exposant le nombre de chiffres après le point) ;

PAS 2 - Tester le premier caractère de la zone ;

si = blanc - mettre zéro dans C ;

- aller à PAS 8 ;

si = + - aller à PAS 3 ;

si = - - mettre - 1 dans le switch ;

- aller à PAS 3 ;

PAS 3 - Tester le caractère suivant ;

si = . - mettre 1 dans B ;

- aller à PAS 3 ;

si = blanc ou si c'est le dernier caractère, aller à PAS 5 ;

sinon, aller à PAS 4 ;

PAS 4 - multiplier A par 10 ;

- ajouter le caractère à A ;

- multiplier B par 10 ;

- aller à PAS 3 ;

PAS 5 - tester B

si B = zéro, aller à PAS 7 ;

sinon aller à PAS 6 ;

./...

PAS 6 - diviser A par B ;  
- mettre le résultat de la division dans C ;  
- aller à PAS 8 ;

PAS 7 - mettre A dans C ;

PAS 8 - multiplier C par le switch ;  
- fin du sous-programme.



## 6. CONCLUSION

=====

Le gestionnaire de grilles améliore la communication entre l'utilisateur et son programme d'application. Cet outil est surtout destiné à l'application de mise à jour de la banque de données du simulateur de gestion. Cette application en exploite toutes les possibilités dont les principales sont : les contrôles de fourchettes, l'enchaînement de grilles et l'envoi de messages. Mais il est également accessible à d'autres applications.

Il est prévu d'apporter une extension au gestionnaire de grilles, qui serait de décrire les grilles à partir d'un écran et non plus à l'aide de cartes.

Nous espérons que cette extension apportera une facilité supplémentaire pour l'utilisation de l'outil que nous avons étudié avec intérêt.

A N N E X E S

## INFORMATIQUE

DEST.

ORIG.

OBJET:

N° REF.

DATE:

PAGE:

		LISTE		DES	GENÉRIQUES			
HEAD	USIN	DØVE	SØCØ	TYFP	DEPT	SDVE	MARC	SDCR
FPTØ	BLØC	M-CL	ENTX	DIRG	SKFP	S-AD	S-AU	S-PR
FLPR	SKPR	SKCH	S-VE	PMCL	S-DG	FPUS	CDEP	ECHX
SQA1	SQA2	SQA3	SQSP	SQSV	FLDC	PLIE	DPP	CVEN
MIX	SQN1	SQN2	SQN3	SQN4	SQP1	SQP2	CACH	FPVE
FPDG	SQN5	SQN6	SQN7	DPTU	SDØU	CSPE	ELCØ	



OBJET :

N° REF.

001

INFORMATIQUE

DEST.

DATE :

ORIG.

PAGE :

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
CF X GEN HEAD										HEADER																			
LIBELLE										UNITE																			
XXXXXXXXXXXXXXXXXXXXXXXXXX										CODE LIBELLE																			
										XX XXXXX																			
DATE										DUREE SIM										PAGE - RANGE									
EXPLOIT										EN JOURS										MINI MAXI									
XXXXXXXX										SNNNN										NNNNN NNNNN									
HORIZON										EN										E1 ENS									
DATE DEBUT										FIN										E2 NNNNN NNNNN									
NNNNN										NNNNN										NNNNN									
CODE ACTION X																													



INFORMATIQUE		OBJET:		N° REF.		002	
COST.				DATE:			
CHRG.				PAGE:			

CF	X	GEN	USIN	USINE										PAGE-RANGE																			
USINE N																																	
LIBELLE				ENS.										MINI										MAXI									
XXXXXXXXXXXXXXXXXXXXXXXXXXXX				E3										NNNNNN										NNNNNN									
</																																	

CF X GEN DØVE		DØMAINE DE VENTE					
DØMAINE XX			UNITE CØURANTE CØDE LIBELLE XX XXXXXX	UNITE GLØBALE CØDE LIBELLE XX XXXXXX			
CØDE ACTION X							















# INFORMATIQUE

OBJET:

N. REF.

007

DEST.

DATE:

ORIG.

PAGE:

CF X GEN SDVE

SØUS-DØMAINE DE VENTE

DØMAINE XX SØUS-DØMAINE X

LIBELLE

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

UNITE CØURANTE  
CØDE LIBELLE  
XX XXXXX

UNITE GLOBALE  
CØDE LIBELLE  
XX XXXXX

CØDE ACTION X











INFORMATIQUE		OBJET:		N REF.		010	
DEST.				DATE:			
ORIG.				PAGE:			

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
CF X GEN FPTØ																													
TYPE FØRN PRIM XX										NATURE XXX										FØURNISSEUR XXXXXXXX									
LIBELLE										UNITE																			
XXXXXX										CODE										LIBELLE									
XXXXXX										XX										XXXXXX									
DISPØNIBILITES										DATE																			
MINI										SAISIE																			
SNNNNNNNN • NNNNN										NNNNNN										NNNNNN									
CODE ACTION X																													



INFORMATIQUE		OBJET:		N° REF. 011	
DEST.				DATE:	
ORIG.				PAGE:	

CF X GEN BLOC										BLOC																			
USINE N										DEPARTEMENT N										BLOC NN									
LIBELLE										UNITE COURANTE										UNITE GLOBALE									
XXXXXXXXXXXXXXXXXXXXXXXXXXXX										CODE LIBELLE										CODE LIBELLE									
										XX XXXXXX										XX XXXXXX									























INFORMATIQUE		OBJET:		N° REF.		018	
DEST.				DATE:			
ORIG.				PAGE:			

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0																						
CF X		GEN S-AU		SECTION AUXILIAIRE																																															
USINE N		DEPARTEMENT N										SECTION XXX																																							
LIBELLE		UNITE D'OEUVRE										TYPE																																							
XXXXXX		CODE LIBELLE										S-AU																																							
XXXXXX		XX XXXXX										N																																							
CAPACITE		EN U.Ø.										DATE										CØUT U.Ø.										DATE																			
MINI		MAXI										SAISIE										SOUS-TRAITEE										CØEF																			
SNNNNNNN		SNNNNNNN										NNNNN										SNNNNNNN										SNNNNNNN																			
VRAISEMBLANCE		PRIX										REVIENT										S-AUX										EQUIVAL																			
MINI		MAXI										GEN.										CØDE										DUREE																			
SNNNNNNN		SNNNNNNN										SNNNNNNN										S-AU										XXXXXX										SNNNN									
CØDE		ACTION										X																																							







INFORMATIQUE		OBJET:		N REF.		020	
DEST.				DATE:			
ORIG.				PAGE:			

CF X	GEN	FLPR	FLUX DE PRODUIT		TYPE		INDIC		CUMUL		CØEF		CHANG	
USINE N			DEPARTEMENT N	BLOC NN	SECTION XXX		FLUX		U.C.		U.G.		UNITE	
LIBELLE			UNITE		LIBELLE		INDIC		U.C.		U.G.		CHANG	
XXXXXX			CØDE		XXXXXX		N		M		N		NNNNNN.NNNNN	
NIVEAU			DATE		SAISIE		INDIC		EDIT.		N		N	
MINI			SAISIE		NNNNNN		N		N		N		N	
SNNNNNNNN.NNNNN			NNNNNN		NNNNNN		NNNNNN		NNNNNN		NNNNNN		NNNNNN	
VRAISEMBLANCE			PRODUCTION		VRAISEMBLANCE		PRIX		REVIENT		MAXI		NNNNNN	
MINI			MAXI		MINI		NNNNNN		NNNNNN		NNNNNN		NNNNNN	
SNNNNNNNN.NNNNN			SNNNNNNNN.NNNNN		SNNNNNNNN.NNNNN		SNNNNNN		SNNNNNN		SNNNNNN		SNNNNNN	
CØDE			ACTION		X									



INFORMATIQUE		OBJET:		N° REF. 021	
DEST.				DATE:	
ORIG.				PAGE:	

CF X GEN SKPR										STOCK DE PRØDUIT										PRØDUIT XXXXXXXX																													
USINE N DEPARTEMENT N BLOC NN										FILIERE XX																																							
LIBELLE										UNITE										INDIC CUMUL										CØEF CHANG										INDIC									
XXXXXXXXXXXXXXXXXXXXXXXXXXXX										CØDE LIBELLE										U.C. U.G.										UNITE										EDIT									
XXXXXXXXXXXXXXXXXXXXXXXXXXXX										XX XXXXXX										N										NNNNNN.NNNNN										N									
PRIX ØRDRE										CØEF										DATE										STØCK INITIAL										DATE									
VALEUR										EVØL										SAISIE										VALEUR										SAISIE									
SNNNNNNNN.NNNNN										SNN.NNN										NNNNN										SNNNNNNNN.NNNNN										NNNNN									
PRIX ØRDRE										CØEF										DATE										STØCK FINAL SØUHAITE										DATE									
CALC										EVØL										SAISIE										MAXI										SAISIE									
SNNNNNNNN.NNNNN										SNN.NNN										NNNNN										SNNNNNNNN.NNNNN										NNNNN									
CØDE ACTION X																																																	







INFORMATIQUE		OBJET:		N° REF. 023	
DEST.				DATE:	
ORIG.				PAGE:	

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
CF X GEN S-VE										SECTION VENTE																			
USINE 0 DEPARTEMENT 9										SECTION XXX																			
LIBELLE										UNITE D'OEUVRE																			
XXXXXXXXXXXXXXXXXXXXXXXXXXXX										CODE LIBELLE																			
										XX XXXXXX																			
DISPONIBILITE										DATE																			
MINI										SAISIE																			
SNNNNNNNN.NNNNN										NNNNNN																			
VRAISEMBLANCE										DUREE																			
MINI										JOURS																			
SNNNNNNNN.NNNNN										SNNNN																			
CODE ACTION X																													



INFORMATIQUE		OBJET:		N° REF. 024	
DEST.				DATE:	
ORIG.				PAGE:	

12345678901234567890123456789012345678901234567890	CF X GEN PMCL	PRØUIT-MARCHE-CLIENT		1234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890	DØMAINE XX SS-DØM X	DØM-CRIT X	AUTRES CRIT XXX	MARCH-CLT xxxxxxxx
12345678901234567890123456789012345678901234567890	LIBELLE	UNITE	INDIC CUMUL	CØEF. CHANG
12345678901234567890123456789012345678901234567890	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	CØDE LIBELLE	U.C. U.G.	UNITE
12345678901234567890123456789012345678901234567890	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	YX XXXXXX	N	NNNNNA.NNNNNN
12345678901234567890123456789012345678901234567890	PRIX VENTE	CØEF	CONTRAINTE MARCHÉ	DATE
12345678901234567890123456789012345678901234567890	SNNNNNNNNN.NNNNN	EVØL	MINI	SAISIE
12345678901234567890123456789012345678901234567890	SNNNNNNNNN.NNNNN	SNNN.NNN	SNNNNNNNNN.NNNNN	NNNNN
12345678901234567890123456789012345678901234567890	VRAISEMBLANCE PRIX	VENTE	CØEF	DATE
12345678901234567890123456789012345678901234567890	MINI	MAXI	EVØL	SAISIE
12345678901234567890123456789012345678901234567890	SNNNNNNNNN.NNNNN	SNNNNNNNNN.NNNNN	SNNN.NNN	NNNNN
12345678901234567890123456789012345678901234567890	CØDE ACTION X			











INFORMATIQUE		OBJET:		N° RÉF. 027	
DEST.				DATE:	
ORIG.				PAGE:	

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0										
CF X GEN CDEP																																							
USINE N										DEPART 1 N										BLOC NN																			
FILIERE XX										DEPART 2 N										PRØDUIT XXXXXXXX																			
LIBELLE										UNITE										INDIC CUMUL										CØEF. CHANGT									
XXXXXXX										CØDE LIBELLE										U. C. U. G.										UNITE									
XXXXXXX										XX XXXXXX										N N										NNNNN. NNNNN									
PRIX CESSIØN										CØEF EVØL										DATE SAISIE										QTE A CEDER/APPRØ EN MØNØ - USINE									
NNNNNNNN. NNNNN										SNNN. NNN										NNNNNN										SNNNNNNNN. NNNNN									
QTE A CEDER/APPRØ EN MULTI - USINE										DATE SAISIE										DATE SAISIE										DATE									
NNNNNNNN. NNNNN										SNNNNNNNN. NNNNN										NNNNNN										NNNNNN									
CØDE ACTION X																																							











INFORMATIQUE		OBJET:		N° REF.		030	
DEST.				DATE:			
ORIG.				PAGE:			

CF	X	GEN	SQA2	STANDARD	CONS	SECT	AUX	PAR	SECT	AUX	DEL
S-AU	:	USINE	N	DEPARTEMENT	N	SECTION	XXX				
S-AU	:	USINE	N	DEPARTEMENT	N	SECTION	XXX				
TYPE RELATION XX											
STANDARD	CONSUMATION			CØEF	EVØL	DATE	VRAISEMBLANCE	STANDARD	MAXI		DEL
SNNNNNNNN	NNNN			SNNN	NNN	NNNN	NNNN	NNNN	NNNN	NNNN	X
CØDE ACTION X											



















INFORMATIQUE		OBJET:		N REF.		035	
DEST.				DATE:			
ORIG.				PAGE:			

12345678901234567890123456789012345678901234567890123456789012345678901234567890	CF X	GEN	PLIE	STANDARD	CONS	FLUX	PRØDUIT	PAR	STØCK	PRØDUIT	12345678901234567890123456789012345678901234567890
	FLPR :	USINE	N	DEPARTEMENT	N	TYPE	FØURN	PRIM	XX		
	SKPR :	USINE	N	DEPARTEMENT	N	BLØC	NN	FILIERE	XX	PRØDUIT	XXXXXX
12345678901234567890123456789012345678901234567890123456789012345678901234567890	PRØPORT	DATE	SAISIE	INDIC	INDIC	CUMUL	CØEF.CHANG.	UNITE	STØCK	CØEF.CHANG.	INDIC
	SNNN.NNN	NNNNN	NNNNN	EDIT	U.C.	U-G.	NNNNNN.NNNNN	NNNNNN.NNNNN	NNNNNN	UNITE	CALCUL
				X	N	N				BLØC	X
12345678901234567890123456789012345678901234567890123456789012345678901234567890	VRAISEMBLANCE	MINI	MAXI	PRØPORTION							
	SNNNNNNNN.NNNNN	NNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNN						
12345678901234567890123456789012345678901234567890123456789012345678901234567890	CODE	ACTION		X							







INFORMATIQUE		OBJET :		N REF. 037	
DEST.				DATE :	
ORIG.				PAGE :	

12345678901234567890123456789012345678901234567890	CF X	GEN	CVEN	STANDARD	CONS :	FLUX	PRØDUIT	PAR	STØCK	ECHANGE	1234567890123456789012345678901234567890
	FLPR :	USINE	N	DEPARTEMENT	N	TYPE	FØURN	PRIM	XX	NATURE	XXX
	SKCH :	JØMAINE	XX	SS-DØM	X	DØM-CRIT	X	AUTRES	CRIT	XXX	ØRIG/DEST XXXXXX
12345678901234567890123456789012345678901234567890	QTE A VENDRE	MØD	CØMPLØT	DATE	SAISIE						
	MINI	MAXI	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN						
12345678901234567890123456789012345678901234567890	QTE A VENDRE	MØD	INCOMPLØT	DATE	SAISIE	HYPØTH	PRØDUIT	PRIX	REV	CØEF	DATE
	MINI	MAXI	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	SAISIE	VENDU	NNNN	NNNN	EVØL	SAISIE
	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN	NNNNNNNNNN
12345678901234567890123456789012345678901234567890	CØDE	ACTION X									







INFORMATIQUE		OBJET:		N RÉF.		039	
DEST.				DATE:			
ORIG.				PAGE:			

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0																														
CF X GEN SQN1																																																											
STANDARD CØNS :										FØURN PRIM PAR SECT ADMINISTR																																																	
SKFP : USINE N										TYPE FØURN PRIM XX										NATURE XXX																																							
S-AD : USINE N										DEPARTEMENT N										SECTION XXX																																							
TYPE RELATION XX																																																											
STANDARD CØNSØMMATION										CØEF EVØL										DATE SAISIE										VRAISEMBLANCE STANDARD										DEL																			
SNNNNNNNN.NNNNN										SNN.NNN										NNNNN										SNNNNNNNN.NNNNN										NNNNN										X									
CODE ACTION X																																																											
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0																														























INFORMATIQUE		OBJET:		N REF.		045	
DEST.				DATE:			
ORIG.				PAGE:			

CF X	GEN	CACH	STANDARD CØNS :		STØCK	ECHANGE	PAR	FLUX	PRØDUIT	
SKCH :	DØMAINE	XX	SS-DØM	X	DØM-CRIT	X	AUTRES	CRIT	XXX	ØRIG/DEST XXXXXX
FLPR :	USINE	N	DEPARTEMENT	N	TYPE	FØURN	PRIM	XX	NATURE	XXX
QTE A VENDRE	NØD	INCØMPLØT			DATE					
MINI		MAXI			SAISIE					
SNNNNNNNN.NNNNN	SNNNNNNNN	SNNNNNNNN.NNNNN			NNNNNN					
QTE A VENDRE	NØD	CØMPLØT			DATE					
MINI		MAXI			SAISIE					
SNNNNNNNN.NNNNN	SNNNNNNNN	SNNNNNNNN.NNNNN			NNNNNN					
CØDE ACTION X										



















INFORMATIQUE		OBJET:		N° RÉF. 050	
DEST.				DATE:	
ORIG.				PAGE:	

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
CF X GEN SQNZ																													
F-DG : USINE 1																													
S-DG : USINE 1																													
TYPE RELATION XX																													
STANDARD										CØEF										DATE									
CONSO MMATION										EVØL										SAISIE									
SNNNNNNNN.NNNNN										SNNN.NNN										NNNNN									
VRAISEMBLANCE										STANDARD										DEL									
MINI										MAXI																			
SNNNNNNNN.NNNNN										SNNNNNNNN.NNNNN										X									
CODE ACTION X																													



















BUMP



0 0 3 5 9 0 0 6 2

\*FM B16/1980/03

